

## HESEA-ACE.

Title	Author	Source
6502 Disassembler		ACELO
Greg Menke	.....	11/00/84 n
65C02		ACELO
Greg Menke	.....	11/00/84 n
830 Modem Auto-Answer		HACK
Bob Crowell	.....	10/00/84 n
ATR-8000 and Terminals		JACC
James Miller	.....	12/00/84 n
Atariwriter Features, More Undocumented		HAUG
Levin C. Soule	.....	11/00/84 n
Atariwriter Underground (part 1)		JACC
Frank Pazel	.....	10/00/84 n
Atariwriter Underground (part 11)		JACC
Frank Pazel	.....	11/00/84 n
B Graph (Business Applications)		PACE
Steve James	.....	12/00/84 n
Buggy Basic		STARFLEET
Trent Ludley	.....	10/00/84 n
Commodore vs Atari (Good, Bad, and Ugly)		Current Notes
Tim Kilby	.....	12/00/84 n
Compuserve, how to get the Most out of		PACE
Martha Dycus	.....	11/00/84 n
DT-80 Cartridge for Atari/ATR System		Current Notes
Bob Danson	.....	12/00/84 n
Lark Crystal help		C Squad
Editor	.....	11/00/84 n
Decimal & Hex Codes for Instruction Set		ANTIC
Antic	.....	12/00/84 n
✚ Enchanter, the solution		C Squad
Editor	.....	11/00/84 n
Enhanced Density for 1050, Rana & Indus		MAAUG
John Reuschlein	.....	11/00/84 n
Eprom Cartridge, Make your own		MAAUG
Bakki Kudva Robert Badeau	.....	11/00/84 n
✚ Lpson FX-80		MAAUG
Stephen McMillan	.....	11/00/84 n
✚ Gemini 10-X		MAAUG
Dave Divine	.....	11/00/84 n
Infinite Precision Division		JACC
Kenneth J. Pietrucha	.....	12/00/84 n
Investment Software		ABACUS
Michael Stomp	.....	12/00/84 n
LED Control by Port		AACE
Rick Lettelsen	.....	12/00/84 n
✚ Letter Perfect V 6		HACK
Bob Rodrigues	.....	10/00/84 n
✚ Letter Perfect, Quick Reference chart		PACEworld
Tim Brown	.....	12/00/84 n
✚ Mule, helpful hints		C Squad
Scott Subeck	.....	11/00/84 n
✚ Musical Notes (part 7)		Current Notes
Jay Gerber	.....	12/00/84 n

## HESEA-ACE

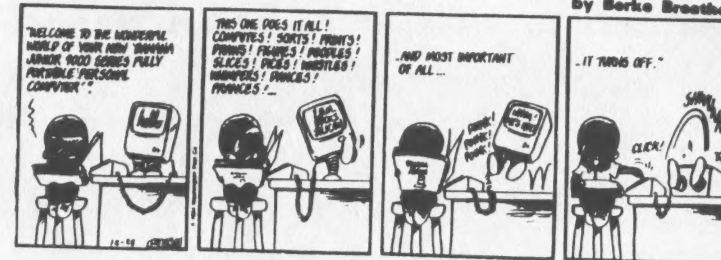
Title		Source
Author	Issue	
Programming Languages, A Guide to	JACG	
Arthur Leyenberger	12/00/84	n
Programming Languages, A Guide to	JACG	
Arthur Leyenberger	11/00/84	n
* Rasterman Vibrations	PACE	
Bard Ermentrout	11/00/84	n
Telecom (Compuserve Service & Prof. Rev)	JACG	
Lawrence Moriano	12/00/84	n
* Tricky Input	STARFLEET	
Trent Ludley	10/00/84	n
* Ulysses and the Golden Fleece, Tips	Current Notes	
Barry Burke Jim Stevenson	12/00/84	n

\* out of frequency to save paper

## by Berke Breathed



## by Berke Breathed



## 65C02 AC6EO

Optimized Systems has available a CMOS version of the 6502 microprocessor, called a 65C02, which can directly replace the normal 6502 in an Atari computer. This chip requires less power, thereby running a lot cooler, and for the assembly language programmer, it offers 27 new OP codes. Currently, however, MACROS is the only assembler to support these codes unless you write your own Macro, or include it via a BYTE command. Here are the new commands with a brief description of each.

**BRA** —branch always This instruction works like all the other branch instructions, except it always branches, and is therefore like a JMP but taking up one less byte of memory and one less cycle to execute.

**INA & DEA** Increment and decrement the Accumulator. Works the same as INX, DEX, INY, & DEY.

**PHX, PHY, PLX, PLY** These instructions work like the PLA and PHA instructions, only pushing the respective register instead of the Accumulator.

**STZ** This stores a zero into the following location but doesn't affect any register. Address modes available are ABSOLUTE, ABS.X, (ZERO PAGE), (ZPG.X).

**TRB** This complements the Accumulator, AND's it with the specified memory location, and stores the results in the memory location.

**TSB** This OR's the Accumulator with the memory location, and stores the results in the memory location. Both TRB & TSB use only ABSOLUTE and ZEROPAGE addressing.

**JMP (ABSOLUTE X)** This instruction takes the absolute address, adds the X register, and jumps to that location. It is a very powerful way of setting up a table of JMP addresses, which are then indexed through the X register.

Additionally the BIT instruction has two new addressing modes: ABSOLUTE.X and ZEROPAGE.X.

There is also a very useful new addressing mode. A common assembly language instruction sequence is LDA #0.

The new addressing mode is OPCODE (zero page). The Y register is not used, but one can use the indirect mode as if Y was set to zero. The following instructions can be used with this mode: ADC, AND, CMP, EOR, LDA, ORA, SBC, STA.

## 6502 DISASSEMBLER

One thing which can really help a machine language programmer is a disassembler to print out his or somebody else's code. If you don't have one, it is not worth \$30 to buy a disassembler limited to listing code from memory or disk file only. A disassembler may sound easy to make, but it isn't.

There are 56 different 6502 ML instructions, some have only one form, but most have up to four different forms performing up to four different things. Because of the number of possibilities, it isn't practical to use upwards of 120 different IF THEN statements.

In my disassembler, rather than using ten million IF THENs, I used a big string subscripted into 4 byte chunks holding the 3 letter ML instruction and an option byte. The option byte tells the program what form to use in the disassembly. The disassembler looks up the instruction by taking the number it is trying to disassemble, multiplying it by 4, and subtracting 3. This gives the instruction's location in the string. The rest is easy, decode the option byte, get any other bytes needed to complete the form of the instruction, and print it.

Type in the program, SAVE it and RUN it. The program will set up the string and prompt you from where to disassemble. If you choose the disk option, a disk directory will come up. Then enter a BINARY LOAD file name. Take care to see the # of sectors in the file isn't greater than the available buffer space, or when it loads, the program will probably crash. If you choose the memory option, enter the start address in hex or decimal. If you use hex, precede the address with a \$.

Next, choose where the output will go. If you choose the printer, be sure it is ready BEFORE you press P. If you type D, enter a filename and hit return. The resultant file will be text, but it will be pretty long if your source code is good sized.

When the program is disassembling, pressing any key will pause it. Then press START to exit back to the start of the program. Press SELECT to continue the listing, or press OPTION to end.

— Greg Menke

## GREG MENKE

## disk File disassembler

```

10 REM Memory/Disk file disassembler
11 REM
12 REM By Greg Menke
13 REM
14 REM 9/9/84 WZ.O
15 REM
16 REM
17 REM
18 REM HEX and Decimal conversion
19 REM routines by Shane Rabin.
20 REM See the October 1983 ACE.
21 REM
22 REM
23 REM
24 REM
25 REM
26 REM
27 REM
28 REM
29 REM
30 REM
31 REM
32 REM
33 REM
34 REM
35 REM
36 REM
37 REM
38 REM
39 REM
40 REM
41 REM
42 REM
43 REM
44 REM
45 REM
46 REM
47 REM
48 REM
49 REM
50 REM
51 REM
52 REM
53 REM
54 REM
55 REM
56 REM
57 REM
58 REM
59 REM
60 REM
61 REM
62 REM
63 REM
64 REM
65 REM
66 REM
67 REM
68 REM
69 REM
70 REM
71 REM
72 REM
73 REM
74 REM
75 REM
76 REM
77 REM
78 REM
79 REM
80 REM
81 REM
82 REM
83 REM
84 REM
85 REM
86 REM
87 REM
88 REM
89 REM
90 REM
91 REM
92 REM
93 REM
94 REM
95 REM
96 REM
97 REM
98 REM
99 REM
100 REM
101 REM
102 REM
103 REM
104 REM
105 REM
106 REM
107 REM
108 REM
109 REM
110 REM
111 REM
112 REM
113 REM
114 REM
115 REM
116 REM
117 REM
118 REM
119 REM
120 REM
121 REM
122 REM
123 REM
124 REM
125 REM
126 REM
127 REM
128 REM
129 REM
130 REM
131 REM
132 REM
133 REM
134 REM
135 REM
136 REM
137 REM
138 REM
139 REM
140 REM
141 REM
142 REM
143 REM
144 REM
145 REM
146 REM
147 REM
148 REM
149 REM
150 REM
151 REM
152 REM
153 REM
154 REM
155 REM
156 REM
157 REM
158 REM
159 REM
160 REM
161 REM
162 REM
163 REM
164 REM
165 REM
166 REM
167 REM
168 REM
169 REM
170 REM
171 REM
172 REM
173 REM
174 REM
175 REM
176 REM
177 REM
178 REM
179 REM
180 REM
181 REM
182 REM
183 REM
184 REM
185 REM
186 REM
187 REM
188 REM
189 REM
190 REM
191 REM
192 REM
193 REM
194 REM
195 REM
196 REM
197 REM
198 REM
199 REM
200 REM
201 REM
202 REM
203 REM
204 REM
205 REM
206 REM
207 REM
208 REM
209 REM
210 REM
211 REM
212 REM
213 REM
214 REM
215 REM
216 REM
217 REM
218 REM
219 REM
220 REM
221 REM
222 REM
223 REM
224 REM
225 REM
226 REM
227 REM
228 REM
229 REM
230 REM
231 REM
232 REM
233 REM
234 REM
235 REM
236 REM
237 REM
238 REM
239 REM
240 REM
241 REM
242 REM
243 REM
244 REM
245 REM
246 REM
247 REM
248 REM
249 REM
250 REM
251 REM
252 REM
253 REM
254 REM
255 REM
256 REM
257 REM
258 REM
259 REM
260 REM
261 REM
262 REM
263 REM
264 REM
265 REM
266 REM
267 REM
268 REM
269 REM
270 REM
271 REM
272 REM
273 REM
274 REM
275 REM
276 REM
277 REM
278 REM
279 REM
280 REM
281 REM
282 REM
283 REM
284 REM
285 REM
286 REM
287 REM
288 REM
289 REM
290 REM
291 REM
292 REM
293 REM
294 REM
295 REM
296 REM
297 REM
298 REM
299 REM
300 REM
301 REM
302 REM
303 REM
304 REM
305 REM
306 REM
307 REM
308 REM
309 REM
310 REM
311 REM
312 REM
313 REM
314 REM
315 REM
316 REM
317 REM
318 REM
319 REM
320 REM
321 REM
322 REM
323 REM
324 REM
325 REM
326 REM
327 REM
328 REM
329 REM
330 REM
331 REM
332 REM
333 REM
334 REM
335 REM
336 REM
337 REM
338 REM
339 REM
340 REM
341 REM
342 REM
343 REM
344 REM
345 REM
346 REM
347 REM
348 REM
349 REM
350 REM
351 REM
352 REM
353 REM
354 REM
355 REM
356 REM
357 REM
358 REM
359 REM
360 REM
361 REM
362 REM
363 REM
364 REM
365 REM
366 REM
367 REM
368 REM
369 REM
370 REM
371 REM
372 REM
373 REM
374 REM
375 REM
376 REM
377 REM
378 REM
379 REM
380 REM
381 REM
382 REM
383 REM
384 REM
385 REM
386 REM
387 REM
388 REM
389 REM
390 REM
391 REM
392 REM
393 REM
394 REM
395 REM
396 REM
397 REM
398 REM
399 REM
400 REM
401 REM
402 REM
403 REM
404 REM
405 REM
406 REM
407 REM
408 REM
409 REM
410 REM
411 REM
412 REM
413 REM
414 REM
415 REM
416 REM
417 REM
418 REM
419 REM
420 REM
421 REM
422 REM
423 REM
424 REM
425 REM
426 REM
427 REM
428 REM
429 REM
430 REM
431 REM
432 REM
433 REM
434 REM
435 REM
436 REM
437 REM
438 REM
439 REM
440 REM
441 REM
442 REM
443 REM
444 REM
445 REM
446 REM
447 REM
448 REM
449 REM
450 REM
451 REM
452 REM
453 REM
454 REM
455 REM
456 REM
457 REM
458 REM
459 REM
460 REM
461 REM
462 REM
463 REM
464 REM
465 REM
466 REM
467 REM
468 REM
469 REM
470 REM
471 REM
472 REM
473 REM
474 REM
475 REM
476 REM
477 REM
478 REM
479 REM
480 REM
481 REM
482 REM
483 REM
484 REM
485 REM
486 REM
487 REM
488 REM
489 REM
490 REM
491 REM
492 REM
493 REM
494 REM
495 REM
496 REM
497 REM
498 REM
499 REM
500 REM
501 REM
502 REM
503 REM
504 REM
505 REM
506 REM
507 REM
508 REM
509 REM
510 REM
511 REM
512 REM
513 REM
514 REM
515 REM
516 REM
517 REM
518 REM
519 REM
520 REM
521 REM
522 REM
523 REM
524 REM
525 REM
526 REM
527 REM
528 REM
529 REM
530 REM
531 REM
532 REM
533 REM
534 REM
535 REM
536 REM
537 REM
538 REM
539 REM
540 REM
541 REM
542 REM
543 REM
544 REM
545 REM
546 REM
547 REM
548 REM
549 REM
550 REM
551 REM
552 REM
553 REM
554 REM
555 REM
556 REM
557 REM
558 REM
559 REM
560 REM
561 REM
562 REM
563 REM
564 REM
565 REM
566 REM
567 REM
568 REM
569 REM
570 REM
571 REM
572 REM
573 REM
574 REM
575 REM
576 REM
577 REM
578 REM
579 REM
580 REM
581 REM
582 REM
583 REM
584 REM
585 REM
586 REM
587 REM
588 REM
589 REM
590 REM
591 REM
592 REM
593 REM
594 REM
595 REM
596 REM
597 REM
598 REM
599 REM
600 REM
601 REM
602 REM
603 REM
604 REM
605 REM
606 REM
607 REM
608 REM
609 REM
610 REM
611 REM
612 REM
613 REM
614 REM
615 REM
616 REM
617 REM
618 REM
619 REM
620 REM
621 REM
622 REM
623 REM
624 REM
625 REM
626 REM
627 REM
628 REM
629 REM
630 REM
631 REM
632 REM
633 REM
634 REM
635 REM
636 REM
637 REM
638 REM
639 REM
640 REM
641 REM
642 REM
643 REM
644 REM
645 REM
646 REM
647 REM
648 REM
649 REM
650 REM
651 REM
652 REM
653 REM
654 REM
655 REM
656 REM
657 REM
658 REM
659 REM
660 REM
661 REM
662 REM
663 REM
664 REM
665 REM
666 REM
667 REM
668 REM
669 REM
670 REM
671 REM
672 REM
673 REM
674 REM
675 REM
676 REM
677 REM
678 REM
679 REM
680 REM
681 REM
682 REM
683 REM
684 REM
685 REM
686 REM
687 REM
688 REM
689 REM
690 REM
691 REM
692 REM
693 REM
694 REM
695 REM
696 REM
697 REM
698 REM
699 REM
700 REM
701 REM
702 REM
703 REM
704 REM
705 REM
706 REM
707 REM
708 REM
709 REM
710 REM
711 REM
712 REM
713 REM
714 REM
715 REM
716 REM
717 REM
718 REM
719 REM
720 REM
721 REM
722 REM
723 REM
724 REM
725 REM
726 REM
727 REM
728 REM
729 REM
730 REM
731 REM
732 REM
733 REM
734 REM
735 REM
736 REM
737 REM
738 REM
739 REM
740 REM
741 REM
742 REM
743 REM
744 REM
745 REM
746 REM
747 REM
748 REM
749 REM
750 REM
751 REM
752 REM
753 REM
754 REM
755 REM
756 REM
757 REM
758 REM
759 REM
760 REM
761 REM
762 REM
763 REM
764 REM
765 REM
766 REM
767 REM
768 REM
769 REM
770 REM
771 REM
772 REM
773 REM
774 REM
775 REM
776 REM
777 REM
778 REM
779 REM
780 REM
781 REM
782 REM
783 REM
784 REM
785 REM
786 REM
787 REM
788 REM
789 REM
790 REM
791 REM
792 REM
793 REM
794 REM
795 REM
796 REM
797 REM
798 REM
799 REM
800 REM
801 REM
802 REM
803 REM
804 REM
805 REM
806 REM
807 REM
808 REM
809 REM
810 REM
811 REM
812 REM
813 REM
814 REM
815 REM
816 REM
817 REM
818 REM
819 REM
820 REM
821 REM
822 REM
823 REM
824 REM
825 REM
826 REM
827 REM
828 REM
829 REM
830 REM
831 REM
832 REM
833 REM
834 REM
835 REM
836 REM
837 REM
838 REM
839 REM
840 REM
841 REM
842 REM
843 REM
844 REM
845 REM
846 REM
847 REM
848 REM
849 REM
850 REM
851 REM
852 REM
853 REM
854 REM
855 REM
856 REM
857 REM
858 REM
859 REM
860 REM
861 REM
862 REM
863 REM
864 REM
865 REM
866 REM
867 REM
868 REM
869 REM
870 REM
871 REM
872 REM
873 REM
874 REM
875 REM
876 REM
877 REM
878 REM
879 REM
880 REM
881 REM
882 REM
883 REM
884 REM
885 REM
886 REM
887 REM
888 REM
889 REM
890 REM
891 REM
892 REM
893 REM
894 REM
895 REM
896 REM
897 REM
898 REM
899 REM
900 REM
901 REM
902 REM
903 REM
904 REM
905 REM
906 REM
907 REM
908 REM
909 REM
910 REM
911 REM
912 REM
913 REM
914 REM
915 REM
916 REM
917 REM
918 REM
919 REM
920 REM
921 REM
922 REM
923 REM
924 REM
925 REM
926 REM
927 REM
928 REM
929 REM
930 REM
931 REM
932 REM
933 REM
934 REM
935 REM
936 REM
937 REM
938 REM
939 REM
940 REM
941 REM
942 REM
943 REM
944 REM
945 REM
946 REM
947 REM
948 REM
949 REM
950 REM
951 REM
952 REM
953 REM
954 REM
955 REM
956 REM
957 REM
958 REM
959 REM
960 REM
961 REM
962 REM
963 REM
964 REM
965 REM
966 REM
967 REM
968 REM
969 REM
970 REM
971 REM
972 REM
973 REM
974 REM
975 REM
976 REM
977 REM
978 REM
979 REM
980 REM
981 REM
982 REM
983 REM
984 REM
985 REM
986 REM
987 REM
988 REM
989 REM
990 REM
991 REM
992 REM
993 REM
994 REM
995 REM
996 REM
997 REM
998 REM
999 REM

```

## by Bob Crowell

This was my overview of the finished project: I would use a regular telephone, but the handset would be "permanently" in the muffs of the modem.

6,104,10,205,6,204,38  
 4030 0070 705,165,204,24,101,00,133,20  
 6,165,205,101,07,133,207,6,204,10,205,  
 6,204,10,205,165,204,24  
 4030 0070 101,204,133,206,165,205,101,  
 207,133,207,165,206,24,101,203,133,206,  
 141,133,6,165,207,105,0,133  
 4040 0070 207,141,134,6,104,133,213,10  
 6,133,212,104,104,141,129,6,206,129,6,  
 104,104,141,131,6  
 4050 0070 165,0,141,132,6,141,130,6,16  
 7,0,141,130,6,172,130,6,177,212,16,5,2  
 06,120,6,41,127,20,32  
 4060 0070 176,5,24,105,46,16,7,201,96,  
 176,1,56,23,32,133,204,169,0,133,205,  
 133,204,6,204,30  
 4070 0070 705,6,204,10,205,6,204,10,70  
 5,165,205,24,107,204,7,133,205,164,200,  
 177,204,77,120,6,172  
 4075 0070 132,6,145,206  
 4080 0070 210,206,165,206,201,0,240,15  
 165,206,74,105,40,133,206,144,227,230,  
 207,200,223  
 4090 0070 144,160,230,132,6,230,130,6,  
 206,129,6,40,43,173,132,6,205,151,6,20  
 7,127,169,0,141,137,6,24,173,132,6,  
 4100 0070 105,64,141,129,6,173,134,6,1  
 05,1,141,134,6,173,133,6,133,206,173,1  
 34,6,133,207,24,144,200,96





NOT on the hookswitch of the phone itself. Therefore, the only way to "pick up" the phone would be to connect the phone to the phone line, as could be done with a switch or relay. I would have to build some sort of a circuit which would work in conjunction with the acoustic modem to handle the chores of INTELLIGENTLY "picking up" and "hanging up" the phone. (7)

Here, then, is the way I tackled the requirements of my gadget, in the order I listed them above...

1. RECOGNIZE THE FACT THAT THE PHONE IS RINGING: Going by a motto that I like, K.I.S.S., or "Keep It Simple, Stupid!", I took the quick-and-dirty approach. I had an old telephone set, picked up at a flea market a long time ago, so I cannibalized it. I attached a small reed switch (like those used in alarm systems) to the coil of the bell. That is, I TAPED the reed switch to the outside of the bell's coil, making NO electrical contact with the phone's wiring at all. The two wires from the reed switch were routed out of the phone casing, for eventual connection to my gadget. When the phone rings and the bell is energized, the coil throws off enough stray magnetism to close the leaves inside the glass-encased reed switch. So, now I had a switch that would close whenever the phone rang. Step one, completed.

2. PICK UP THE PHONE: Now that I had the switch above, it seemed to be a simple matter to answer the ringing line. I would connect a small relay in series with a power supply and that reed switch above, and whenever the phone rang, the relay would be activated, and I could wire the contacts of the relay to connect the phone line to the talk circuits of the phone. The PROBLEM would be KEEPING the phone picked up! Obviously, when the phone rang, my Line relay would pull in, answering the phone and stopping the ring, but immediately, since the phone wasn't ringing anymore, the relay would open again, and it would hang up the phone. (A VERY short call!) What I needed was a DELAY-type relay, one that would STAY energized for a short time after being triggered, perhaps for 15 seconds or so. Well, years ago, I had purchased a surplus relay of that type, but when I unearthed it from my junk box, it turned out to be defective (So much for surplus!). However, in the process of searching, I found a motorized darkroom timer, one that when triggered, would keep an internal relay closed for an adjustable amount of time up to 15 seconds. Perfect! So, I did some testing, and found it to work well. I connected the reed switch on the bell coil to the trigger contacts on the timer, and when the phone rang, the timer's relay closed, the timer motor started up, timed 15 seconds, and then the relay contacts opened again, hanging up the phone. Well! I was ahead of myself... I had not only found a way to answer the phone, but KEEP it answered for a short period, which would be necessary for part of #4, below!

3. TURN ON THE ANSWER-MODE CARRIER: Easy! I would LEAVE the modem switched to the Answer mode, and just use extra contacts of the Line relay in #2 above, to turn on the power to the modem. Solved.

4. WAIT A PREDETERMINED TIME FOR A RETURN (ORIGINATE) CARRIER. Okay. Already done:

A. IF NO RETURN CARRIER, HANG UP AND RESET: The timer in #2 above would keep the phone line "off-hook" for up to 15 seconds, and then hang up.

B. IF A CARRIER IS RETURNED BY THE ORIGINATING CALLER, KEEP THE PHONE LINE "OFF-HOOK" AND LET THE BBS PROGRAM TAKE OVER: This is where it gets a little more tricky... Like I mentioned above, the B30 modem has a signal line which is energized when there is a carrier present. (Actually, I feel somewhat obligated to note that it's slightly more involved than that, electronically, but for the purposes of this discussion, I'll leave it like that.) I decided that all I needed to do was connect a small low-current relay to this line, and this new relay, in turn, could keep the Line relay energized. In case I just lost some of you, here's a review of what we've got so far:

The ringing phone line triggers the timer/delay relay, which, in turn, turns on the Line relay, and that "picks up" the phone. If a carrier gets returned to the modem by the caller's modem before the timer "hangs up", the small carrier detect relay is also energized, and this continues to energize

the Line relay after the timer gets done, so this KEEPS the phone "off-hook" as long as a carrier is present. Got it?

5. CONTINUOUSLY MONITOR THE LINE TO SEE IF THE CARRIER IS LOST, (8) AND IF SO, IMMEDIATELY HANG UP AND RESET: Also solved, since if carrier is lost at any time, the Carrier-Detect relay would open, causing the Line relay to open, putting the line back "on-hook". Also, the BBS program, monitoring the RS-232C signal lines, would also note the loss of carrier, and reset the program to wait for another caller.

So, with the theory of operation all planned out and taken care of, all I had to do was assemble everything I needed, find some way to mount all the various parts, and wire the thing together. I must tell you, when I got done with the first "prototype" of my gadget, it looked like an electronic technician's nightmare!

I used a piece of plywood about one foot square to mount the power supply (a 110 to 24 volt AC stepdown transformer and bridge rectifier) on, along with the "Line" relay and the small "Carrier-Detect" relay, and connected wires from these individual parts to separate terminals on a screw-terminal barrier contact strip so it would be easy to re-wire things, if necessary. The "darkroom" timer had its own screw terminal strip on the back, and was sort of free-standing, so I didn't mount it on the board. I slightly modified the wiring in my cannibalized phone so that the bell would be always connected to the phone line, but the "talk" circuits could be switched on and off. (Actually, I just fixed it so the contacts of my Line relay replaced those in the standard phone hook-switch.)

I had to somehow get the two wires from the CRX (Carrier Detect) signal out of the modem, and after a little investigating, discovered the easiest way would be to simply open the plug on the cable from the B50 interface to the B30 modem (at the B30 end, a DB-25 male connector), and tap onto the terminals therein. That way, I didn't have to open up or disturb my modem at all. Neat.

Well, I wired it all up, carefully checked for any short-circuits, and then began to test. Everything seemed to be in order, so with just about every finger and toe crossed, I connected this hodge-podge to my Atari system, plugged it in, and guess what? It worked perfectly, first time! Not only was I elated, but AMAZED! I just couldn't believe that I'd actually done it! I had a couple of people call and test it out for me, and then the next day I put up a special version of the BBS program during the day, and I tested it myself, from my office at work. Amazing... smooth operating, all the way.

Several days after this initial test, I decided that if I wanted to KEEP this gizmo for any real usage, I would have to clean up the "packaging"; it looked AWFUL: a phone that looked like it had EXPLODED from within, a piece of wood with all sorts of loose parts and wires sticking up, and a modem with an open plug. It covered about three square feet of floor space! I eventually re-mounted all the components in a more logical order on a different piece of wood, slightly smaller and neater, and tidied up my wiring. Although it's no work of art, it sure looks a lot better! With the addition of a couple of plugs and jacks, it's now possible to separate the gadget from the B30 modem, which was not possible earlier, since it was all "hard-wired" together. I also added a push button to simulate the phone ringing, so I can take the phone "off-hook" at will, and I added a power switch and fuses for safety. It doesn't have an enclosure or cover yet, but I'm working on it...

I only had to make one change to the BBS software to be compatible with this gadget, and that was the "Good-bye" routine. Since my gadget will ONLY hang up the phone when it loses the carrier from the remote caller, the problem became apparent that if a caller logged off with the "G" command, BUT stayed on the line, the gadget would never lose carrier, and so would dutifully keep the phone off the hook, while the BBS software would reset, and immediately, it would think that there was a new caller on the line, and attempt to log it on. A minor change in the program will take care of that.

If you read my article last month regarding my experiences with the assembly of a "kit" B90 computer, and my concluding remark, "Now, if I only

knew what to use it for...", you will see that my two "semi-home-brew" projects would make a great team, and I could have my own BBS up and running! Now, if I only had another disk drive... Hmm... I wonder...  
Stay tuned!

Bob

>>>> LETTER PERFECT, V.6 <<<< 9

Rod Rodrigues

Some of you might have peeked at the new version of letter perfect; some of you might be contemplating it, wondering if it is worth making the move. In a word, it is; I say this in spite of the fact that the deletion commands and the deletion functions commands have been changed. Because of this, if you are going to make the change, you should just put the old one away and make the transition to the new one. Let's go on to why it is worth it.

Letter Perfect is in this writer's opinion the best and most powerful word processor for the Atari. Those of you who started with it years ago will remember when you couldn't use it, except on Atari printers. They came out with a "patch" type program called a disk printer editor, and a new version of Letter Perfect. To use this combination, you had to load the DFE, use it to type in your own codes for the printer controls, and save the printer driver to your new data disk. Then at boot up time, you were given a user unfriendly EADQ menu. To load your personalized printer driver from your data disk, you had to insert your data disk, type D, then the # sign. Finally, the main menu appeared on the screen.

Then you might have gotten Spell Perfect to check the spelling of your documents. To do so, you had to save your document, reboot the new program, then reload your file, then insert the dictionary, then do some more disk swapping after you were through correcting the spelling. It was a frustrating experience. In one fell swoop, Letter Perfect has, in the version 6, made preparation and spell checking much easier.

In doing so, they have moved to the type of method used on the IBM's when you first get a word processor for it. You must use an INSTALL program to configure the program for your system. Some call the install program a configurator, as a result. To configure you basically have to tell the program which printer you are using and which type of monitor. In other words, you do the job we used to do with the Disk Printer Editor. Only now, you will be creating an autorun program disk which knows at bootup time which printer you have.

To configure your system, make sure you have your printer codes handy. Press the ESC key with LP v.6 in the drive. You are then going to be led through a series of screens telling the computer what kind of screen (40 or 80 column) you have, how many drives you have, which drive will be the file drive, and which drive will be the dictionary drive, etc. At one point, you will load one of the printer drivers. Load the one which looks closest to your own. You will then be asked if you want to edit the driver. If so, you will be led through all the screens as you used to in the DFE. At the end of the edit time, you will be given several options. I have used the option to save the driver and add it to the directory of available drivers. That way, if I am not happy with the one I got, I can later reload it in the same way and edit it. Note that you will at that point be saving it to the program disk itself. After you have saved it, you are returned to the main menu. I am told that you will have to reboot after that in order to activate the printer driver. I have had no trouble having the printer driver take over at that point already. At any rate, it is good practice to reboot anyway.

Now you are in for a treat. XL owners will not need the translator any longer. Just boot, unless you have the BOOXL, in which case, boot while holding down the OPTION key down. The system will already be configured for your screen and your drive configuration; the printer driver will also be loaded already. The first thing you see will be the main menu. It is different from the old one. First of all, the option it lands on is LOAD. Some people like that.

HACI

MORE UNDOCUMENTED ATARI WRITER FEATURES

By Levin C. Soule

HAUS  
NOV 84

The July 1984, ACE of Syracuse, NY., had an article on one use of OPTION INSERT (stop printing and insert text from keyboard) and CTRL V (chain file). The procedure will allow you to save a form letter and later add information as prompted by the program. It was carried in our August 1984 newsletter. I was able to get the INSERT function to work, but not the CTRL-V function, at least not as written in our newsletter. Through trial and error I now know how to do it and have found this to be a very powerful function. Atariwriter is a complete mail merge and/or data merge program all in one.

The first thing to do is to type a form letter. !!!!! At each point in the letter where you will later make an insert, hold down the OPTION key and press the INSERT key. The future insert can be anything from zero to 35 characters long. Follow the insert command with either a SPACE or a RETURN as, or if, needed. This is all in chapter 4, page 6 of the instruction book. Save your letter. Now comes the fun part.

Create a new file. Delete the format line. Type in the first =insert (35 characters max.). Press RETURN. Type in the second insert. Press RETURN. And so on until all inserts for ONE COPY of the form letter are complete. Printer commands can be included and can cause the insert line to exceed the 35 character limit. However, the code must be complete within each insert line. You can string two inserts together in the middle of a WORD if you insert two insert commands without a space between them. Now repeat with the next set of inserts for the next copy of that same form letter. Continue until you have your complete set of inserts. One important thing to remember is that your complete letter and complete set of inserts is all going to have to fit in memory at the same time. Therefore you may have to make several sets of inserts and save them under different file names, such as INSERT1, INSERT2, INSERT3. If your inserts are addresses or the like, you can use the insert file with any letter or document you want.

Now we will do a mail merge. Load your letter. I will assume that you have a list of 25 names and addresses to merge with 25 copies of your letter. Select the Print Option from the Menu. When asked how many copies, enter at least 25. I suggest using more, as the program will just stop when you run out of inserts and wait for you to make a keyboard entry. When the program starts printing and comes to the first insert, it will stop and ask for an insert. For disk, enter CTRL-V D:FILENAME.EXT (NOTE: THE CTRL-V WILL NOT SHOW ON THE SCREEN. AND DO NOT FORGET THE D: BEFORE THE FILE NAME). Include drive # if needed. For cassette enter CTRL-V C:. Press RETURN. If your list is so large that you must use several insert files, enter enough copies at the start to cover all needed copies. Then when the program stops, enter CTRL-V D: and the new insert file name.

By using pin feed file cards and changing the number of lines per page to fit the length of the file card, you could use this method to type file cards for say the PTA. This would also be a cheap way for a small business to send out individualized dun letters, to include typing the envelopes. It can also type mailing labels. If you have a letter quality printer without tractors, just put a CTRL-W in the format line, set CTRL-Y to 12 or some multiple of 12 (for 1" labels or 18 for 1 1/2" labels). For letters just use CTRL-W. This will stop the printer to allow you to check alignment of the labels or put in a new sheet of paper.

I wonder what other undocumented things ATARIWRITER can do? About three days after I wrote this article I received the Oct JACC Newsletter and Frank Pazel had Part One of an article that had this same info in it. So I will wait and see what other things ATARIWRITER will do, and report in later newsletters.

JACK OCT 84

# Atariwriter Underground Part I by Frank Pazel - JACK

## The Form Letter

The good folks at the former Atari Inc. gave us what is probably the best word processor for any PC, bar none. It is easy to learn, handles just about any kind of request, is loaded with features, and is absolutely dirt cheap. With the advent of the APX Printer Driver almost any printer works with it and the use of AtSpell is a godsend to those less lexical. It is, in short, superb.

Now, these very same departed programmers had a few tricks up their collective sleeves when they put this ROM together. They weren't entirely honest in reporting all the well surprises packaged inside this 5 cubic inch marvel. Or perhaps they told all but between technical writer and manufacturing much was lost. In any event, in two parts I am going to try to report what I have gathered from various sources and have discovered through experimentation about the oft demeaned Atariwriter. (Who can forget the slap in the face from Time magazine's "Man-Computer of the Year" issue when they said "Atari's word processor is not for the serious user.") Shame. Stupidity.

Hidden inside the cartridge lies the latent ability to create form letters automatically with a mail merge, block copy text from one file to another, and unleash a resident modem handler. This month, the mail merge.

There have been a couple of articles written on this feature in other newsletters but following their instructions led to frustration and no results. It turns out that a key point was always left out that I stumbled on almost by accident. Here is my report on how to get your Atariwriter to function as a bonafide business type form letter producer.

A form letter, for definition purposes, is a document which will have certain parts of it contain personalized information. The bulk of the letter is the same for each addressee. You get these things in the mail every week telling you how you might just have won 18 billion dollars. Suppose you have the need to produce such a letter. Perhaps your club needs to send out a mailing which would be nice to look personalized, or might attract more attention if the receiver's name appears inside the text. Using normal Atariwriter functions write the letter. However, whenever you want to personalize the text hold down the OPTION key and press the INSERT key at the same time. An inverse ESCape character will be printed on the screen in that position. Later on, when you ask Atariwriter to PRINT your document the program and printer would normally halt and allow you to type in the missing information. This procedure is detailed on page 39 of the Atariwriter manual. As the manual says, "...leave blanks in a text file....and fill them in each time you print the file." The

underlining is mine. This is exactly what we want to avoid. We want to create a file which will automatically merge with our letter file and insert the missing information for us.

When you are sure your letter is exactly like you want it SAVE it. Make accurate notes of how many blank items you need to fill in and what the information needs to be. For example, the first three ESCape characters might represent name, street address, and town and state. Create a new file in the following manner:

1. Enter the Editor and delete the entire format line. Yes. You should now be looking at a blank blue page which used to have inverse letters with numbers after them. This is the absolutely crucial step in making this process work.

2. Using your notes about the empty blanks in your letter type in the missing information with a RETURN at the end of each piece of information. Use no blank lines and continue typing in your repeating series of data. You are creating a "sort of" data base for the letter. Hopefully, you will design it so it can be used for other things. The addresses, for example, can be used to make up labels later on.

3. SAVE this file. I use the name MERGE but you use your favorite. Count the number of records in this file. A record is all of the information you need to print one letter. It might be something like name, street, and city-state. The number of records will equal the number of different documents you are going to print. You have ten different names with addresses in your MERGE file; you are going to print ten different personalized letters.

4. LOAD your letter into Atariwriter. Turn on your printer. Position your paper. Begin the PRINT series. At the prompt "PRINT WHOLE DOCUMENT?" answer Y.

5. At the prompt "NUMBER OF COPIES" type in the number of letters you are going to print. This number should equal the number of records in your MERGE file. The maximum is 99.

6. At the prompt "MAKE ENTRY, PRESS RETURN", hold down the Control key and press V (CTRL-V) and you should hear the one key click. Now type in the specifications of your data file, e.g., DIMERGE. Make sure your MERGE file disk is in your active drive.

7. As soon as you press RETURN the printer should come to life and begin churning out your form letters. If you have specified right hand justification (J1) each letter will be printed with the personalized information properly justified. It's near magic!

Some of this information comes by the way of the San Leandro (CA) Computer Club which reports a mysterious interoffice memo at Atari briefly outlined its existence and use. Who knows? I collected bits and pieces here and there, from newsletters, talking to people, and just plain dead-end trying. The procedure works. And it makes what is the most versatile and valuable PC glitter on another facet of its diamond-studded personality.

NEXT MONTH: Block text moves between files and modem to modem.

## Current Notes

## MUSICAL NOTES

Jay Barber (part 7)

I hope you found last month's dissertation on music history of interest. This month, I will review and explain the workings of, along with some tips on getting the most out of, *Advanced Basic System II*, from LotsaBytes.

A much respected institution among Atari computer owners was the Atari Program Exchange, or API. This company thought up an ingenious idea: to let the 'common folk' who use the Atari, design and program software and market it at prices substantially less than commercial retail. Out of this effort came an abundance of high-quality, low-cost programs dealing with entertainment, home management, education, and, with the work of the very talented Lee Actor, *Advanced Basic System II* (ABS).

The program that Lee designed was a remarkable package that would edit and play virtually any piece of sheet music on the Atari computer. Up until that time, the only music editor for the Atari was the *Music Composer* cartridge, put out by Atari, Inc., at the same time they introduced their 400/800 computers. ABS was far better than *Music Composer*, as it allowed a full 6 octave note range, compared to *Music Composer*'s 3. It could also deal with complexities not possible in Atari's cartridge like triplets, notes faster than 64ths, and accents. In fact, it was the most advanced music package for any computer under \$2000.

Since then, the *Advanced Basic System* has become the most-used music package on the Atari computer. You will be hard-pressed to find an Atari User Group anywhere that does not have some ABS song files in its library. With the advent of Actor's *Advanced Basic System II* (ABS II), creating music on the Atari is easier than ever. Editing commands have been speeded up, and you can now enter a piece as large as an entire movement from a symphony.

To give you an idea of how easy and extensive ABS II's capabilities are, I'll go through a sample session of using it, explaining and pointing out the best features of the program.

First, we pick out a piece of sheet music we really like, say Mozart's Eine Kleine Nachtmusik. We then boot up ABS II, and see the main menu. From this menu, we select "B" for entering music. Then ABS II asks us which of the four voices to edit. Let's choose 1, to type in the main melody. A window appears on top of the screen, telling us:

- The voice number (out of four),
- The measure number (1, since we are just starting),
- The number of beats in the measure currently being edited,
- The meter, or time signature,
- The tempo, which can be changed anywhere in the entire piece!

- The key signature (C=00, D=20, F=10 where 0=sharp and b=flat),
- The octave number of the current note being edited, from 1 to 8,
- The duration of the current note (w=whole, h=half, q=quarter, etc.),
- The envelope (more on this later),
- The dynamic from ppp to fff and the accent from Fl to F7,
- And the amount of notes that can be added until there is no more memory.

At the bottom of the screen, there is a message that looks like this:

Note[00][Dur][Env][Dyn]

This is the prompt for you to enter the next note. It is also the format in which you enter the notes into ABS II. The first note of Eine Kleine is a G-natural quarter note in the 5th octave. The dynamic marking is fortissimo, or loud. The way you would enter this note is: G5Q/F

So, the pattern in which you enter notes is: name of note, octave number, note duration, envelope, /dynamic marking. A C-sharp sixteenth note in the 3rd octave played double pianissimo (very soft) would be entered: C3S/PP.

If your next note is: in the same octave, of the same duration, or has the same envelope and/or dynamic marking as the previous one, then these commands can be deleted from your entry. For example, the second note (not counting rests) is a D eighth note. Since it is also played fortissimo, and also resides in octave 5, the entry for this note would be simply DE.

By now, you might be wondering about the mysterious "envelope" command I keep mentioning. The envelope tells the computer how to play the note. The default is "s" which is regular length notes. If you want to slur or tie the note to another one, then you would use the envelope "l". There are also commands for staccato (short) and double staccato (very short) notes. These appear in music as a little dot over (not next to) the note.

After you finish entering all the notes in a measure, just type "M" to go on. If you entered too many or too few notes, then the computer will buzz and tell you that you don't have the correct number of beats in the measure. You may take this feature off if you want, in case you are working with irregular measures. You can also change the time signature, or meter, any time you want. However, this does not affect playback, and is only used for editing (i.e. if you started in 4/4 and switched over to 2/2, which is double-time, or twice as fast, you could enter in the notes correctly, but the piece would still play at the original tempo, unless you used the change tempo command.

Like I mentioned before, you may change the tempo anywhere in the piece by entering in "T" followed by the



new tempo number. *AMS II* treats the tempo command like a note that can be edited at any time. Also, you can put the tempo command in any voice, and as many times as you like.

After you finish entering in voice one, then type "0", and the program again displays the main menu. You can either type "B" to continue editing any voice, or choose "A" to listen to what you have typed in. If you don't want to hear the entire piece, then you can choose option "B" to hear any portion or section you desire.

Although you can change the tempo at any time within a piece, you must select a starting tempo with option "H". The tempo number directly matches the standard metronome markings of a piece. Usually you will find the markings just above the first line of music. It will read: (picture of quarter note)=111. The 111 is the number which you enter in for the tempo command.

The rest of the options deal with input/output functions such as: disk directory, load file, save file, delete file, format disk, and, when you want to get rid of a piece in memory to start another, the "C" or erase voices option, which will clear all memory, and reset the tempo to the default value (100).

The best features of *AMS II* are its editing features. As I mentioned before, *AMS II* will accept any duration of note that exists in music. Triplets are handled with a "3" after the duration, as in E3 for a eighth note triplet or S3 for a sixteenth note triplet. A septuplet or any n-tuplet for that matter is handled in the same way. If these conventional durations fail to please you, you can enter any note that you can possibly imagine by using a ratio of "play x many notes in the time it takes to play y notes".

The measure and note editing commands are just as extensive in scope. To repeat a note or series of notes, you use the command REPx,y. This will repeat the first x notes y times. The command for measures is REPMx,y or repeat measures x thru y and place them in either the current measure (if empty), or the following measures. To insert a measure, just type in 1. Also, if you want to move a block of measures from one place in a piece to another, you can use the REPM command on the measure right before you want the block inserted.

As you can see, *AMS II* is a remarkably well-designed music package. With all its features and ease of entering, I would have to say that *Advanced Music System II*, by Lee Actor, is the best music package for any computer, bar none.

*Advanced Music System II*, by Lee Actor. Marketed by LotsaBytes, 15445 Ventura Blvd., Suite 106; Sherman Oaks, CA 91413 for \$14.95 plus \$1.95 shipping/handling on 24K disk and 16K cassette.

## JACK MECHY Terminals and the ATR

by James Miller - JACG

The ATR8888 with the Atari as a terminal in CP/M allows you to use great software. Wordstar, dBase III, and SuperCalc I find are good mostly because they are in 80 columns. The DT-88 cart is a lot easier to read than Atrmon88 but to go one step better, I'm using a Lear Siegler ADM 3A Video Display Terminal. The book that came with the ATR doesn't say much about using terminals. So the following may be helpful.

Back of the terminal is an RS-232 port.

terminal to Atari 1/0 13 pin Jack

pin 2 (send)	pin 3 (data in)
pin 3 (receive)	pin 5 (data out)
pin 7 (ground)	pin 4 or 6 (ground)
	pin 1 jumper to pin 10

You must set jumpers J7 and J10 inside the ATR for terminal use. J7 is a double so with J10 that's 3 jumpers to change. Fine if you want to stay in CP/M mode. Radio Shack has a 3 pole switch # 275-661 I use (wire wrapped not soldered!) to these jumpers so I can go back and forth between Atari control and CP/M without opening up the ATR each time to set the jumpers. Also so I don't have to unplug the terminal when I go back to Atari control, I added a 2 pole switch on the jump of Atari jack pin 1 + 10 and between ADM pin 3 to Atari pin 5. Now both Atari and terminal may be left connected at all times to the ATR8888. Some terminals (like Perkin Elmer) look for data set ready signal and need its RS-232 pin 4 jumped to pin 5, and pin 4 jumped to pin 20 before they come to life. Don't forget to set the terminals to 9600 baud, odd parity and 7 bits. Now you have the best of both worlds (CP/M and Atari) all in one system, and clear displays. Now then, where did I put that DOS disk ...



"SHOOT! MY JACG MEMBERSHIP RAN OUT!"

## Atariwriter Underground Part II

by Frank Pazel - JACG

Phone Lines and Block Moves

The Atariwriter ROM has, in addition to the ability to do a form of mail merge discussed in last month's newsletter, the mechanism to transfer files via a modem.

In order to use this hidden modem handler you must boot up a copy of the original DOS 2.0 Master Diskette which came with your disk drive. Most people are unaware that stuck away on its mysterious recesses is the RS232 information for handling modem operations. If you are using OSS software it is a file called "RS232.COM". Both communicators must be running through an 850 Interface Module. Using Option E rename it "AUTORUN.SYS" and you are in business. Once both ends of the telephone connection have contacted each other files are SAVED or LOADED from "Rfilename". Try it, and save some transfer time.

The final little trick that Atariwriter will do for you is a variation on its Duplicating Text feature. Rather than using the copy function to copy within a file you can use it just as well to copy from file to file. Use the Duplicating Text sequence described on page 37 of the instruction manual. This amounts to marking the beginning and ending of the text block you want to move with a CTRL-X. At this point, however, press ESC and return to the menu. Select to Create a new file or Load a file, depending on how you want to use your extracted block of text. If you load a file enter the Editor, position the cursor where you want to enter the saved block of text and press OPTION D. The saved file has been residing in the copy file buffer and can be used again and again. This is especially handy if you are preparing a report which uses a special format that must be repeated. To repeat copy just place the cursor where you want to replicate the saved block of text and press OPTION D. No need to remark and save it each time. If you save a new block of text with CTRL-X, that new text will, of course, replace the previously saved block.

The Atariwriter is truly a fine piece of software. Each day I wonder how I could get through my workweek without it. If you discover some new or undocumented features please send them along so we can publish them for the good of the order.

Thanks for some of the source material for this article goes to Clyde Pritchard of the Portland Atari Club and an article in the ACE of Syracuse newsletter.

**"Being a computer means  
never having to say you're  
sorry."**

Art Buchwald  
(1925- )  
Laid Back in Washington



Whenever I get into something that has more options and variables than I can handle I make myself a handy reference sheet that I can go to instead of carryin' the book around. The authors try to make their reference sheets too, but they are usually too big. This one is made to be cut out and stuck on a 5" by 8" file card. I usually put it right above the number keys so it is between the keyboard and the source document I am typing from, or the monitor screen. If you don't have your own yet, I hope you can use this one.

**FRONT LINE OPTIONS**

font.....f  
left margin.....m  
line width.....w  
indent.....i  
neg. indent.....a  
top margin.....t  
bottom margin.....b  
line spacing.....l  
lines per page...p  
justify.....j  
page numbering...s  
stop printing...s

**PRINTER CHARACTERS**

Very special print character...CTRL V  
Boldface toggle.....CTRL B  
Center next line.....CTRL C  
Footer.....CTRL G  
Forced end of page.....CTRL P  
Format line.....CTRL F  
Header.....CTRL H  
Marker at cursor.....CTRL D  
Underline toggle.....CTRL U  
Forced end of line.....RETURN  
Page number.....3

**FUNCTIONS**

Continuous scroll.....CTRL 4  
Scroll 1 page fwd.....CTRL 0  
Scroll 1 page bwd.....CTRL ;  
Fix window width.....CTRL 3  
Rejustify text.....CTRL J  
Kill type ahead.....ATARI  
Quit editor.....ESC  
Block line right.....CTRL C CTRL C  
Left/right footer.....CTRL G CTRL G  
Left/right header.....CTRL H CTRL G

**SPECIAL PRINT CHARACTERS**

(Preceded by CTRL V)  
CHR 8 function.....( )  
Conditional page break.....X  
Data base number a.....0a  
Data base string a.....0a  
File link....."  
Halt printer.....!  
Subscript.....(down arrow)  
Superscript.....(up arrow)

**DELETE FUNCTIONS**

(Preceded by CTRL K)  
All after cursor...A  
All before cursor...B  
Copy buffer.....C  
Up to marker.....H  
All text (now).....H  
To end of Para....R  
All tabs.....T

**DELETIONS**

Character left.....Del Backsp  
Character right.....CTRL Del Backsp  
Current Video Line.....CTRL 5  
Line from cursor...Shift Del Backsp  
Word to left.....CTRL L  
Word to right.....CTRL H  
Delete functions.....CTRL K

**TEXT MOVES**

Copy to buffer.....CTRL 7  
Copy from buffer.....CTRL 9  
Move to buffer.....CTRL M  
Get text from buffer.....CTRL 8  
Save to buffer.....CTRL /  
Lift from buffer.....CTRL L

**CURSOR MOVES**

Beginning of line.....CTRL A  
End of line.....CTRL Z  
Jump to marker.....CTRL X  
Advance paragraph.....CTRL 6  
Bottom of page.....CTRL 0  
Top of page.....CTRL CLEAR  
Next page.....CTRL 0  
Previous page.....CTRL 0  
Begin of text SHIFT CLEAR  
End of text.....CTRL E  
Word left.....CTRL W  
Word right.....CTRL Y

**SEARCH AND REPLACE**

Search.....CTRL S  
Replace.....CTRL R  
Global replace...CTRL CAPS

**INSERTIONS**

Character at cursor.....CTRL INSERT  
Continuous insert mode.....CTRL I  
Line at cursor.....SHIFT INSERT

**TABS**

Clear tab at cursor...CTRL TAB  
Reset default tabs.....CTRL 2  
Set tab at cursor...SHIFT TAB  
Tab to next tab stop.....TAB  
Show tabs.....CTRL T

**BUSINESS APPLICATIONS FOR THE ATARI****B GRAPH**

By Steve James

Beginning this month, I will try and write a regular column about using the Atari computers for "business" applications. By "business" I mean tasks normally associated with the office environment, such as budgeting, accounting, bookkeeping, filing, word processing, and analyzing data. However, since we all do many, if not all of these tasks at home, I will from time to time discuss applications more suited to home rather than office use.

It seems that "business" computing means only one thing to most people -- IBM. While the IBM-PC (and its numerous clones) make excellent microcomputers for the office environment, you can often get close to the same results with less expensive computers. The Atari computers have commonly been perceived as capable only as good game machines and the management teams at Atari have done little to change the general public's viewpoint. However, like any full featured microcomputer, the Atari machines are inherently capable of much more. True, there are limits to their utility and they can not match the IBM-PC in terms of software availability or hardware performance but, the Atari computers can often function satisfactorily for small tasks, for small businesses, or for personal and home applications.

One of the best examples of how well you can use the Atari for "real" computing is the business/scientific graphics program called B/GRAPH. B/GRAPH is presently being marketed by Batteries Included, a firm that previously marketed only Commodore software. Because B/GRAPH has been handled by two other companies previously, it has not received the attention it deserves. The first version of B/GRAPH (v 1.0) was handled by the Canadian firm INHOME software. In 1983, INHOME went bankrupt and the program authors, Michael Reichmann and Robert Wilson arranged with Commodore for distributing an upgraded version of B/GRAPH. Eventually, Commodore decided not to publish software for other computers and retained the rights to publish Commodore 64 and Plus 4 versions of B/GRAPH. The authors got the rights for Apple and Atari versions back and arranged for Batteries Included to publish version 1.1. Advertisements for B/GRAPH have finally begun to appear in the major Atari magazines.

B/GRAPH would first appear to be just another business package for preparing graphs from data. However, a few things set it apart. First of all, it is for the Atari for which good graphing packages are virtually nonexistent. Secondly, it features a comprehensive statistical package that is well integrated into the program. And finally, it comes with an easy to read and understand manual (which many expensive packages for the IBM-PC lack). The documentation is presented in a tutorial format which makes learning a breeze, even for people not familiar with the theories of graphing and statistics. The program comes on two diskettes and requires the Atari BASIC cartridge.

The disk will selfboot and first presents you with a menu of printers that the program supports directly. It will fully support Epson/Gemini ( & other Epson clones), NEC/C. ITOH/Prowriter, Seikosha/Axiom AT 100, Okidata 92, and Centronics printers. If you own a different type of printer, you can always save your graph image to disk and use a BASIC picture file loader (supplied with B/GRAPH) in conjunction with a screen dump utility that supports your printer to get hardcopy graphs. You are then presented with a main menu that allows you to reset to a different printer, graph data, make pie charts, reimage or label graphs, manipulate the data, or perform a variety of statistical analyses of your data. The program also has a "mini-DOS" for diskette housekeeping. Each of these menu choices is a separate module. Some are on

Disk 1 and some are on Disk 2. The program prompts you when you need to switch disks.

Once in the graph module, you can create and edit graphs from data entered manually or obtained from another module, such as the file manipulation or statistics modules. Since the program actually reloads each module into memory, you must save your data as a disk file whenever you move from one module to another. Fortunately, the program always reminds you that it has data in memory and asks you if you want to save it before exiting a module. It would be more convenient to keep the data in memory, but I suppose the limitations of 40K require that data be stored on disk between modules.

The graph program lets you choose from line, point, simple bar, segmented bar, and market (high-low-close) graphs. With each graph you have several options, such as the type of scaling, visible or invisible grid lines (in both horizontal and vertical directions) and color choices. Once you have created a graph you can quickly switch to another type or change the colors, hues, or intensity of the image. You can also call up your data for editing at any time during the session then go back to the graph. By pressing START you dump a copy of your graph to your printer if it is one that is supported by B/GRAPH. The program limits you to graphs with three factors (i.e. sets of data), but you can superimpose two or more graphs if need to show more than three factors at once. You can save your graph data or "image" the graphics mode 8 screen to disk for use in other modules. The data is saved without a graph and the image is saved without data. To cover all your bases, you must save both data and an image for each graph.

The graph module offers only a few options for labeling and titling the graphs. However, in the Graph Imaging/Labeling module, you can customize each graph to suit your tastes and needs. This section allows you to enter text in one of four sizes and, with version 1.1, use different fonts. B/GRAPH comes with nine custom fonts, including a BASIC program that "rotates" a font so it can be printed sideways. You can use any "normal" font file, such as those created by the font programs that have appeared in ANALOG and ANTIC. Once you finish modifying the graph, you can save its graphics mode 8 screen "image" to disk as a 66 sector (single density) file. This is a special format picture file which can be loaded and displayed with a non-copy protected BASIC program included with B/GRAPH. B/GRAPH also has



routines to create a "slide show" of these images, displaying one after another. Since the display routines are not copy protected, they can be transferred to another disk and the number of images in any slide show is limited only by disk storage space.

Version 1.0 had several bugs, mostly in its routines for saving graph images to disk. As far as I can tell, version 1.1 has corrected these. It also offers a host of new features. Changing colors of a graph is easier. You have the choice of solid or dashed grid lines. You can fill the areas under a line graph by simply pressing the F key while a graph is displayed. While examining or editing your data, you can page forward as well as backward with single keystrokes. Also, the ability to use different fonts, including mixed fonts, was not available on version 1.0.

The statistical features of B/GRAPH are impressive for any microcomputer program. You can do simple sample statistics (mean, standard deviation, variance, skewness, etc.), calculate values for t, F, and Chi-square tests, calculate normal, binomial, and Poisson distribution probabilities. A regression module lets you do least-squares linear regressions of single independent variables. The calculations support equations with any BASIC functions (e.g. SIN, LOG, SQR, etc) and degrees greater than one. Version 1.1 does not support multiple linear regressions, but a fix is on the way.

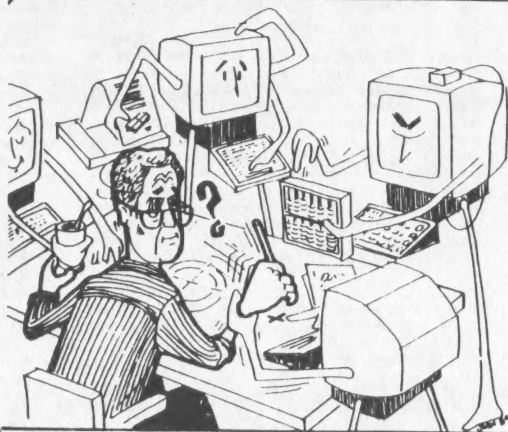
In the File Manipulation module, you can transform, recalculate, and modify your graph data. You can also switch factors if that will make for a better looking presentation. B/GRAPH has routines for accepting data in either ASCII or DIF format and converting it into the B/GRAPH data format. This allows you to input data from other programs such as Visicalc, SynCalc, or SynFile+ directly into B/GRAPH.

If all these features were not enough, the program authors will soon release an Enhancement disk that is used with the program. They anticipated the need to update and add features, so they built in an "Enhancement Disk" selection on the main menu. When the enhancement becomes available, you will get additional printer support and plotter support. A multiple linear regression program will be included to overcome the limitations in the present regression module. Additional graph types, including horizontal bar charts, bubble charts and X-Y charts, will be available as well. These additions came as a result of suggestions by users -- the authors really listen to the complaints and comments of the people who buy their program.

B/GRAPH is simply the most flexible and useful graphing and data analysis program available for a low-cost microcomputer. The excellent graphics abilities of the Atari computers make for great looking graphs. The program is, considering its multitude of features, extremely easy to learn and use. Although it has reasonably flexible labeling functions, you can dress up your graphs by loading the data into a general purpose graphics editor. The most useful program for editing is Picture Plus Version 2 from Non-Standard Magic, PO Box 43, Girard Ohio 44420. This program will load the B/GRAPH images in directly then let you edit and print them in black & white or color in a variety of sizes. Other graphics utilities can be used by converting the B/GRAPH images to a standard graphics mode 8 screen file format by a routine on the B/GRAPH disk. Picture Plus does this directly.

Anyone wishing to use graphs for illustrating data for business, school, or home should get B/GRAPH. It is an excellent tool and proves the Atari computers can do real computing.

#### P.A.C.E. BULLETIN BOARD



COMPUTER-AIDED ENGINEERING

#### TRICKY INPUT

by Trent Dudley

Do you wish you could sometimes omit the ? prompt that appears on the screen when you use an INPUT statement? Unfortunately, the only way to avoid this is to do something like OPEN #1,4,0,"E:" in the initialization part of the program and then use INPUT #1 instead of INPUT, right? Wrong! Due to an undocumented feature (in other words, a bug) in Atari BASIC, using INPUT #16 will trick BASIC into omitting the prompt without having to open another channel. Here's how it works:

When BASIC encounters an INPUT statement, it checks for a channel number; if none is found, as in a plain INPUT statement, channel 0 is used. If channel 0 is to be used, BASIC prints the prompt; since 16 < 0, BASIC decides not to print it. Then BASIC multiplies the channel number by 16, discards any multiples of 256, and checks for a result less than 128, to keep you from trying to use channels 8-15, which don't exist. But 16\*16=256 => 0, so channel 16 passes the test and is treated thereafter as channel 0. Thus INPUT #16 proceeds just as if it were a plain INPUT statement, but no prompt is printed.

Here's another trick you can use to run most machine language programs from BASIC. Using this trick, a menu program could be written which would run both BASIC and machine language files. First OPEN channel #1 to read the file, e.g. OPEN #1,4,0,"D:FILENAME.DOS". Then do X=USR(5576). This is an illegal call into Atari DOS 2.0, so don't try it with any other DOS! If you want the file to load without executing, POKE 5534,255 before doing the USR call (this is like using /N with DOS option L). If you've done the POKE or the file has no init or run addresses, DOS will return control to BASIC, which will generate ERROR- 9. You can ignore it in immediate mode or TRAP it in a program.

#### STARFLIGHT DEF 84

#### BUGGY BASIC

by Trent Dudley

I just read "Another BASIC Bug" in the October issue (#23) of ANALOG Computing and feel I can clear up some misconceptions raised and offer a more satisfactory remedy. The bug described is not a new one, but just another manifestation of a problem with the GET statement described in The Atari BASIC Source Book, page 277.

The bug occurs when a GET statement is executed immediately after a VAL statement. BASIC changes the first character of the string whose VALUE was taken to whatever character the GET statement got. The error is caused by the failure of GET to reinitialize INBUFF, the BASIC buffer pointer located at \$F3. VAL changes INBUFF to point to the string whose value is to be taken. Then GET calls an I/O routine which uses the contents of INBUFF as the CIO buffer address. The failure to reset INBUFF to point to the general purpose BASIC buffer at \$580 before calling CIO causes the byte gotten to be placed in the string that VAL pointed to, instead of the safe buffer area.

To remedy this, you can execute a statement known to reset INBUFF, such as DUMMYS=STR\$(0) or PRINTING a number, before executing GET. A better solution is to initialize INBUFF via the following statement:

X=USR(ADR("hLQZ"))

where Z means inverse Z

This is the step which GET omits.

Incidentally, the INPUT statement doesn't cause this bug since it does properly initialize INBUFF. However, a problem will occur if you are using page 6 for a machine language subroutine and then INPUT (from disk or cassette) a string longer than 128 bytes. Since the string is stored starting at \$580, the excess will spill over into page 6, wiping out the code you had stored there. The second half of page 6 (\$680-\$6FF) is always safe because INPUTs are limited to 255 bytes plus an EOL.



## Commodore vs Atari: The Good, The Bad, and The Ugly

By Tim Kilby



It's the duty of a software developer, given the task of producing software products for the two most popular graphics computers, to try to be objective in assessing the quality of the two hardware systems in question. Having had experience programming Atari computers and being spoiled to certain of its features, I didn't take lightly the request to transport several commercial programs to the Commodore-64. The argument was made that since both used versions of the 6502 microprocessor, reprogramming couldn't be that difficult. But, I remembered conversations with other software developers about the horrors of Commodore programming. Nine months and four conversions later, I'll evaluate for you both systems from a programmer's point of view.

First, let me reiterate a well known fact. More Commodore-64 computers are sold than all other home computers combined. Yet very few home computers are used for programming even though most buyers have that as an objective. If one were buying either of the two computers, Atari 800XL or Commodore-64, based on commercially available programs, it's just about a toss up. Word processors, data base managers, spreadsheets, communications programs, and lots of games are available for each. However, if one's objective is to learn programming by programming, to use BASIC language and perhaps assembly language, to learn and utilize the system's many features in original programs, then one system far exceeds the other.

**Screen Editors.** Let me start with a direct comparison of BASIC screen editors. The Atari editor is a full screen editor just made for programming. Move the cursor past the right margin and it wraps around to the left. Move the cursor past the bottom edge and the cursor appears at the top. There's SHIFT-INSERT to make room between lines for new lines. And SHIFT-DELETE for taking up slack. CTRL-DELETE scrolls a statement backwards, eating up characters as it goes. And TABs and automatically repeating keys speed the editing process. Write statements with or without spaces between words and the screen editor automatically inserts them in the proper places. Perhaps the most important thing for BASIC programmers is instant syntax checking as a program line is entered. The Commodore-64 has none of these features. There is cursor movement in all four directions, but it's not as complete as with Atari. And it is very annoying to not notice syntax errors until a program crashes.

**Graphics.** Commodore BASIC is very similar to Microsoft BASIC and it is built into the system in the form of ROM—just like the 800XL. Commodore BASIC, however, does not have any graphics or sound commands. The hardware supports extraordinary screen displays and sound, but all graphics and sounds are done through PEKs and POKes. Imagine trying to program the equivalent of Atari's DRAWTO command in Commodore BASIC. There are only four Commodore graphic

codes; they are equivalent to Atari's GRAPHICS 0, 8, 12, and 15. The palette of colors is limited to 16 or less, far from Atari's 128.

**Sprites vs Players.** The one outstanding graphics feature which makes Commodore worthwhile is sprites—known on the Atari as players. Commodore can have eight simultaneous sprites, each either single or multi-colored, each 24 bits wide by 21 bytes tall, and each single or double size in both X and Y scales. Sprites are also moved easily in both X and Y directions by a single POKe. They are limited to the 40 X 25 line screen area and may not appear to move into the screen from the outside edges. There can be an almost unlimited number of shapes stored in memory, pointers being changed rapidly for animation effects. Atari, by comparison, is very limited. One nice feature of Atari, however, is that players can be the full height of the screen.

**Sound.** Sound is good on both machines but really very limited to the BASIC programmer. Atari does have the SOUND command which is usable for certain beeps and buzzes. Commodore has a much more sophisticated sound chip which is capable of extraordinary synthesized music. Both machines require machine language speed for anything other than simple sounds.

**Display Lists.** Commodore-64s do not have display lists or display list interrupts, although this type of interrupt can be simulated using raster scan interrupts. Mixed modes can be achieved, multiple character sets used, and dynamic color switches incorporated. Vertical blank interrupts are about the same on both machines, although Atari gives one the choice of two stages for the VBI to occur.

**Scrolling.** One drawback from Commodore's lack of a display list structure is the inability to easily scroll horizontally. Both machines have fine horizontal and vertical scrolling. But since there is no display list in the Commodore, mode lines cannot exceed the standard 40 characters. Thus, it's hard to have objects from a virtual screen scroll into the visible screen. Atari makes that easy by being able to redefine screen memory on any mode line. As with most machines, where there is a will, there is a way. Whatever can be done on the Atari can probably be simulated on the 64, but probably with a lot of extra programming.

**D.O.S.** The disk operating system of the Commodore-64 is adequate but has some design flaws. First, files must be scratched (deleted) before an updated version can be written to the disk with the same filename. And if one is not careful with disk IDs, other valuable files can be overwritten unexpectedly.

(Continued on Page 8)

### Commodore vs Atari (Continued)

A command driven structure—vs. Atari's menu driven DOS—means trying to remember all the commands; they're not displayed on screen. Syntax is not that easy either.

OPEN 3,0,3,"0:filename,S,M":CMD3:LIST 10-90

is the same as Atari's

LIST "0:filename",10,90

By the way, there have been at least two different operating systems that weren't entirely compatible with each other. And Commodore-64s are not compatible with previous Commodore models or their new PLUS model.

**Binary Files.** One good feature is that binary files can be loaded directly from BASIC. On the other hand, one unforgivable design flaw is the lack of anything like an AUTORUN.SYS file. Programs cannot be booted directly from the disk. There's always the LOAD "filename",0 command that must be typed to get a program booted. And for a designer of children's software that believes in total friendliness, this is a definite handicap.

**Copying Disks.** Lastly, there's no disk duplication command and the fastest commercial copiers take about four minutes and four disk swaps to copy a disk. Multiple disk drives apparently were not considered desirable.

**Memory Management.** Hats off to Commodore for excellent memory management. All 64K of RAM is available for programming. BASIC and the operating system (the Kernel) are in ROM and are "transparent" to certain hardware features. For example, character sets and screen data can occupy the same address space as the operating system. And the system can be using both simultaneously. Furthermore, BASIC and the operating system can be switched off and replaced by a user written operating system. Some of these features are in the 800XL but not quite as flexible as with the 64.

**Quality Control.** Hardware reliability is certainly worth considering. At my office we have four Commodore-64 computers and three 1541 disk drives. All of the computers have something wrong: from non-working keys to shorting connectors to outdated video circuits. The disk drives are much slower than even Atari's, if you can believe it.

**Summary.** All in all, do I like Commodore-64? No. Would I buy one for myself? No. Will I continue to program the 64? Yes. After all, I do listen to the market. It's amazing what shrewd, aggressive marketing will do for a product. Atari really missed the boat with what I think is a more "user usable" machine.



### TIPS 'N TRAPS

Barry Burke and Jim Stevenson

Welcome to the second installment of "Tips 'N Traps". Last month we put in some hints for you adventurers out there to solve the ON-LINE adventure *Wizard and the Princess*. We haven't gotten any replies on BBS, mail, or telephone. We are **DESPERATELY** and patiently waiting for some problems or answers to turn up. Without them, this column will soon run out of ideas. We'll first be printing the different problems and answers St. Game originally put out, but since they have gone bankrupt, there is no way we can keep putting in new information to this column without your help.

Also, last month, we told you we were going to give you some addresses to write to. Here they are:

Jim Stevenson-4408 Manor Hall Lane, Fairfax, VA 22033 (703) 378-4093

Barry Burke-12411 Ramrod Court, Fairfax, VA 22030 (703) 830-1978

You can also send us messages on the new **AMRUBIC** message board, "Adventures @ & A." (See the article on the upgraded AMRUBIC elsewhere in this issue. Ed.)

Next month, we'll have some questions, and semi-answers to problems from *The Dark Crystal* and *Mask of the Sun*, and probably some problems and semi-answers from you, the reader/adventurer. Keep those notes and letters coming!

### ULYSSES AND THE GOLDEN FLEECE

- Q. How do I get the Harpies to let me free the man in the cage?
- A. Many men have died in at least one of the Sevenses to find that answer.
- Q. How do I get past Pluto?
- A. Another one bites the dust!
- Q. How do I get past the wall of fire?
- A. The rich aren't the only ones who can bathe in wine.

-or-

- A. Quit your whining, you may get wet.
- Q. How do I get past the fjord in the cavern?
- A. If you remember your mythology, do as Baedalus and Icarus did!
- Q. How do I handle the skeletons?
- A. If I could save time in a bottle  
The first thing that I'd like to do  
Is say the ancient phrases with reversible meanings  
And chop up a skeleton or two.
- Q. How do I get past the dragon?
- A. Diamonds are a dragon's best friend.



## RASTERMAN VIBRATIONS

By Bernd Erentraut

This is the first of a series of articles about ATARI bit mapped graphics. Much of what I will discuss holds true for other computers that use ATARI's simple bit mapping scheme. These articles will not tell you how to use player missile or character graphics. To follow them you should be familiar with binary numbers and some logical operations such as AND, OR, and XOR (exclusive or). The techniques I will use will be described in English and programmed in FORTH and 6502 machine language. BASIC programmers should not despair since it should be easy to follow the code. ACTION! programmers might want to use these algorithms as well since they will run fast in ACTION!

First I will define some simple concepts and describe the layout of bit-mapped graphics. I will also briefly review some logic operations. The second article will describe a fast machine language routine for plotting a single point on the screen. The next article deals with filling a closed area with color. Then, I will describe algorithms for drawing lines and ellipses or circles. The remaining articles will depend on what you guys want to hear about (e.g. rubber lines, anti-aliasing, 3D graphics, animation, etc.)

There are many articles around that can help you understand ATARI graphics. Some of the best are those written by Tom Hudson of ANALOG on many aspects of graphics. A recent article on ATARI graphics modes appeared in the SEPT ANTIC (the one on graphics). If you are puzzled by what I write you should study some of these issues. A good book to start out with is "Computer Animation Primer" by David Fox and Mitchell Waite. This book is for the ATARI. A more advanced text filled with goodies (written in PASCAL) is "Fundamentals of Interactive Computer Graphics" by James Foley and Andries van Dam (the latter authored the graphics article in the SEPT 84 Scientific American).

First I will review some bit logic. As you know, computers only deal in 1's and 0's that is base 2 or binary arithmetic. There are 4 basic logical operations for a pair of bits, b1 and b2. The following table describes them:

b1		1	1	0	0
b2		1	0	1	0
<hr/>					
b1 OR b2		1	1	1	0
b1 AND b2		1	0	0	0
b1 XOR b2		0	1	1	0
NOT b2		0	1	-	-

If you haven't seen this before Tom Hudson has written a series of tutorials ("Boot Camp") in ANALOG which will tell you more. Each byte of information consists of 8 bits. The above operations can be applied to two bytes by applying the rules bitwise to each of the 8 bits. For example let Z1=220 (11100100) and let Z2=41 (00111001) then Z1 AND Z2 = 36 (00100100), Z1 OR Z2 = 253 (11111101), Z1 XOR Z2 = 217 (11001101).

Continued on page 10

Continued from page 9

What are bit-mapped graphics? The graphics modes 3-11 (2-15 on XL except the four color character mode) on the ATARI are all bit-mapped modes. That is each point or pixel on the screen can be individually plotted and colored. For example COLOR 2:PLOT 10,20 will plot a pixel in color 2 at the position 10,20 on the screen. A picture on the screen consists of many such points the information about which lie in the computer's memory. Two important aspects of bit mapped graphics are the resolution (points across by points down) and the total number of available colors. The ATARI can have up to 16 colors on the screen or resolution of 320x192. Our examples will be in GR7x16 with a resolution of 160 across by 96 down and four colors. To understand how the computer plots a point, you must understand how the image on the screen is kept in the computer's memory. This is very easy on the ATARI, hard on the APPLE II and C64 computers. In GR7 each horizontal line on the screen is described by 40 bytes. Each byte contains information for 4 pixels (a pixel is a single point on the screen). So 40x4=160 points across the screen. GR7x16 requires 96 rows of 40 bytes each so that it needs 96x40=3840 bytes. The lowest resolution mode is GR3x16 which needs only 240 bytes but gives a resolution of 40x24. So you trade off memory for resolution. The highest resolution modes (GR24, GR7plus, GTIA) require 40x192=7680 bytes. Suppose that in GR23 (GR7x16) you plot 220 somewhere in the screen memory. What will you see? 4 pixels in a row on the screen will change to a new color (or maybe some will remain the same) You will see colors 3,2,1 and 0 in a row. The reason for this can be seen if you write 220 in binary (11100100) and break it up into groups of two bits, one for each pixel:

11 10 01 00

As you can see, these 4 numbers are the binary representations of 3,2,1,0, the 4 colors seen on the screen. Thus, each byte in GR7 or any other 4 color modes codes for exactly 4 pixels and each pair of bits codes for the colors. In the two color graphics modes, only one bit is needed for each pixel (the pixel is either on or off) so that in these modes each byte codes for 8 pixels (there are 8 bits in a byte). In the GTIA modes such as GR11, there are 16 colors. You need 4 bits to specify 16 colors so that each byte only codes for two pixels. That's why the horizontal resolution in the GTIA modes is so low, 40 bytes per line times two pixels per byte is 80 pixels. Summarizing, there are 2 pixels per byte in the 16 color modes (and GR10), 4 pixels per byte in the 4 color modes, and 8 pixels per byte in the 2 color modes. Pick some random bytes and try to figure out what colors will be displayed in each of the 4 modes.

For example, 123 = 01111011 in binary so in GR7, a 4 color mode you would see ( 1 3 2 3 ) and in GR11 a 16 color mode you would see ( 0111 1011 ) = ( 7 11 ). Try this with 214,255,99,30,4,192,15. I'll give the answers next month.

Now that you understand how each byte in memory is represented on the screen, let's see how to address a single pixel on the screen. Again the example will be in GR23. The coordinates on the screen are (0,0) in the upper left corner and (159,95) in the lower right corner. There are 160x96 pixels on the entire screen. Suppose that we want to plot a point at 23,19 in color 2. How does the computer accomplish this? Basically, it uses the x and y values given to calculate where in the screen memory this point can be found. Then it changes the value of that byte and stores it back into memory. Remember that each byte has 4 slots ( xx xx xx xx ) each consisting of two bits of color information. Let's number them ( 0 1 2 3 ). (In the 2 color modes there are 0 slots and in the 16 color modes there are 2 slots). So, let's plot 23,19 in color 2. Let SCTOP denote the address of the first byte of screen memory. This can be found with 80 0 in FORTH and SCTOP=PEEK(80)+256\*PEEK(00) in BASIC. Since the pixel is in the 19th row and there are 40 bytes per row, the address of the first byte in the 19th row is:

SCTOP+40\*19

Pretty easy isn't it!! Finding the horizontal offset is a little harder. Let's look at the first byte in a given row; it looks like ( p0 p1 p2 p3 ) where each p0,p1,... corresponds to a pair of bits coding for color. The leftmost pair p0 corresponds to the leftmost pixel on the screen, that is the horizontal coordinate is 0. The next pair, p1 corresponds to x=1, and so on until you hit p3. After that you must move to the next byte over to get the x=4 pair of bits. We want the 23rd pixel and every 4 pixels moves us over 1 byte so the 23rd pixel is in the third slot of byte 3 (remember the first byte in a given row is byte 0). That is we write 23 divided by 4 is 5 with a remainder of 3. Hence the byte we want to change in order to plot 23,19 in color 2 is:

SCTOP+19+40\*5

Let's lift this byte out of memory and see what must be done. Since the remainder of 23 divided by 4 is 3 we need to put color 2 in the third slot. Before we change the byte it looks like:

[ xx xx xx \* ]

We want to put color 2 into this slot so that afterwards it looks like

[ xx xx xx 10 ]

since 10 is binary for 2.

Continued on page 11



Continued from page 10

That's all there is to plotting a point:

1. Compute vertical offset
2. Compute horizontal offset
3. Figure out the address of the pixel
4. Figure out which slot to change
5. Put color in and store byte

To see how the computer does it, let's put color 1 at (14,49).

1. vert. offset =  $49 + 40 \times \text{CTOP}$
2. horz. offset =  $(14/4 \times 3) + \text{remainder } 2$
3. address is  $49 + 40 \times \text{CTOP} + 3$
4. we must change slot 2 by putting in color 1.

First the computer erases slot 2 by AND'ing with an OFFMASK 11 11 00 11 which only affects slot 2 (1 and anything else doesn't change it). Next we must insert color 1. This can be done in a number of ways. The easiest is to designate the color 1 by 4 pixels of 1:

[ 01 01 01 01 ]

(Color 2 is 10 10 10 10, Color 3 is 11 11 11 11). Next AND this with a mask that is only on for slot 2:

[ 00 00 11 00 ]

to get [ 00 00 01 00 ]. Take this byte and OR it with the byte we want to change and then store it back in memory. This seems to be a lot of trouble, but next time, I'll include a short little program to do it.

SEE YOU LATER,  
Bard

## BOOK REVIEW: HOW TO GET THE MOST OUT OF COMPUSEVE

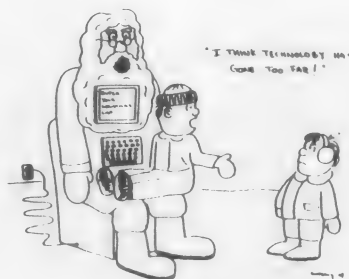
Book by CHARLES BOWEN & DAVID PEYTON

Reviewed by Martha Dycus

This book takes you by the hand and leads you through the jungle of CompuServe with a few chuckles thrown in. It is very easy to read and follow the directions. You are shown the time consuming way of finding what you want by following the menu trail via an on-line tour. Then you are given commands to use that will cut your connect time way down. Did you know that you can set it up so that you are automatically sent directly to the area you choose the moment you log on?

Granted, you can dig out these commands from your CompuServe manual, but "dig" is the correct word. This book is "readable", while a manual usually is not. There is an "On-line Survival Kit" at the end of the book that gives you a quick summary.

To order this book type in "O C18-54 at the first prompt. This will take you directly to CompuServe Product Ordering. From this point you are not charged connect time while in this service. You will be given instructions and then some ordering options. Take #1 "User Guides and Manuals". From this menu pick #5 "Books on Computing". There you will see some choices including "How to Get the Most...\$12.95". The instructions say to allow 4 weeks for delivery, but I received mine in about 1 week.



## The DT-80 Cartridge For the ATARI/ATR System

By Bob Danson



The SMP ATR8000 is a 64K Z80 computer that attaches to an ATARI computer via the ATARI serial data bus. The ATR provides connections for the attachment of generic 5 1/4 and 8-inch disk drives, a parallel printer, and a serial device (which is usually a modem). Code resides in the ATR so that whenever the ATARI computer is being used the ATR functions as a disk drive controller and a peripheral controller (ATARI 850 replacement). But the roles can be reversed with the ATR being the primary computer. In that case the ATR becomes a CP/M system and the ATARI with its attached TV or monitor becomes a terminal from which the ATR is run. In order to do this, a program must first be loaded into the ATARI that will cause it to emulate an ASCII keyboard and a CRT display device. The ATARI keyboard is more than adequate for this function and the TV or monitor can provide the required CRT display - so long as the 40-column by 24-line ATARI Graphics 0 mode is adequate. Unfortunately most CP/M applications such as WordStar, dBase, and SuperCalc were not intended to be used with a 40-column display. They work best with an 80-column display. This means that the ATARI must either have an 80-column board and monitor (the best way to go, but expensive) or the ATARI must generate an 80-column display through a software driven Graphics 8 screen. SMP supplies such a Graphics 8 terminal program for the ATARI - Autoter80. It is a disk-based (a protected un-back-upable disk) that emulates a Lear Siegler ADM-3A terminal. This program does not come with the ATR - it must be purchased for an additional \$29.95 plus \$3.00 shipping. Up to now it was the only such commercially available program.

**The DT-80 Cartridge.** The DT-80 is a cartridge for the ATARI computer that turns the ATARI into a ASCII terminal with an 80-column by 24-line display and allows the ATARI to be used as a terminal for the SMP ATR8000 CP/M system. It provides all the functions of the SMP Autoter80 program plus it has the following additional features:

- The cursor has three modes: flashing, steady, and invisible.
- The background luminance can be changed to one of eight levels.
- The character luminance can be changed to one of eight levels.
- The screen color can be changed.
- The screen display can be set to normal coarse scrolling or to fine smooth scrolling.
- A joystick or trackball can be used as a cursor positioning device and the speed of the cursor movement can be regulated.
- The ASCII codes generated by the keyboard and the joystick can be changed.
- Support for some CRT escape control sequences not provided by Autoter80.

- A menu from which to select and control the DT-80 functions.

The DT-80 program is written by Claus Buchholz and is distributed by ACE - Available Computer Enhancements, P.O. Box 10233, Lansing, MI 48901. The telephone number is 517-351-3092. The cost of the DT-80 is \$39.95, plus \$3.00 for shipping, with an additional \$2.00 for C.B.U.

**The 80-column Display.** Before further discussing the DT-80 a few words must be said about an 80-column display generated in software via an ATARI Graphics 8 screen.

An 80-column Graphics 8 display represents individual characters with a 4 by 8 character dot matrix. A 4 by 8 dot matrix is just barely adequate for good character definition. It can produce a readable display on a black and white TV, a monochrome monitor, or on a color monitor such as the Commodore 1702 when the monitor is used with the ATARI 800 chroma/luminance signals. All these devices work because they don't produce or display artifacts, which is a distortion in the color signal when it is modulated with the luminance signal to produce composite output. Artifacts can be a crippling blow against a Graphics 8 display when the highest possible image quality is required. Because of artifacts an 80-column Graphics 8 display on a composite monitor is less readable, but can be marginally acceptable if the color controls are turned down to minimize the effects of artifacts. A normal color TV usually does not produce an acceptable display.

**The Hardware.** The DT-80 I received came in a grey plastic case. The rear of the case was damaged, as if an old product name or label had been pried off.

The DT-80 cartridge would not fit into my ATARI 800. A careful examination found that the printed circuit board was too wide to fit into the ATARI cartridge slot. After disassembling the case and applying a file to the edges of the circuit board it fit.

It must also be noted that the circuit board has tinned leads, not the better quality and performance gold leads found on the great majority of ATARI cartridges.

All-in-all I found the quality of the DT-80 case and circuit board disappointing.

**The Software.** The software is what the DT-80 was purchased for and there are fewer disappointments here.

The DT-80 produces a typical Graphics 8 80-column display. There are differences in the character set compared to the SMP Autoter80 program, but whether the DT-80 display is overall better or worse is up to the user.

25

The advantage of the DT-80 is that the user can change the display luminance and color. The display can be light with dark characters or dark with light characters. This also allows control of screen contrast without having to adjust the TV or monitor controls. The ability to change the screen color is not quite as useful as the luminance controls, but it can help to produce a more pleasing display when used with a composite color monitor.

The ability to redefine the keyboard is interesting but its use can become tedious: there is no way to save new definitions; they must be entered whenever the DT-80 program is reloaded. One place the keyboard redefinition function is necessary is with the use of a joystick as a cursor positioning device. By default, the DT-80 defines the cursor positioning codes as those used by the Atari - the CTRL arrow keys. But the DT-80 is not currently used as an Atari control program; it's used as a CP/M terminal control program. While there are no standard cursor positioning keys for CP/M, some of the most frequently used keys are those used by programs such as WordStar and dBase - the CTRL-F, S, D, and I keys. Therefore the joystick cannot be used with WordStar unless the cursor positioning codes are redefined. It would be much better if you could save and reinstate alternate cursor positioning codes. Or even better, the WordStar cursor control codes should have been provided as a menu option or for a joystick plugged into the second joystick port.

Another useful DT-80 feature is that many of the display escape control sequences shown in the SMP CP/M Supplement, section V.3, are implemented by the DT-80. Section V.3 has, up to now, been somewhat a work of fantasy, since many of the codes are not ADM-3A codes and were not implemented in AutoLaser. Two such codes are the Line Insert (ESC E) and the Line Delete (ESC R). When they are installed in WordStar they allow WordStar better control over the screen, producing a better display as lines of text are inserted or deleted. I have found that when installing CP/M software, if the install program has the Lear Singler ADM-31 as a choice, the DT-80 will provide the proper terminal characteristics.

The smooth scrolling option of the DT-80 produces a very pleasing effect. As lines are added to the bottom of the screen the display appears to just glide smoothly upward. (But note - this only happens if the CP/M application program treats the Atari as a dumb terminal and allows the DT-80 screen editor to scroll the display. If an application program uses the Line Insert and Line Delete control codes to manage the screen display then the DT-80 smooth scrolling may not work.)

**Performance.** The use of a terminal emulator program from a cartridge is a vast improvement over one on a disk. It is no longer necessary to insert the disk, boot the system, wait for the program to load, and in the case of AutoLaser, hear that loud annoying bee-eep sound. Just plug in the cartridge and you're ready to boot CP/M. If the Atari SYSTEM RESET is pressed there's no need to reload the terminal program - the DT-80 just restarts.

The speed of the DT-80 code is adequate for all its functions. This means that when using the DT-80 no CP/M

software install tricks are needed to get features such as inverse video to work properly.

**In Summary.** It is unfortunate that ACE uses poor quality hardware to package its product. Also, the lack of built-in joystick CTRL-E, S, D, and I cursor positioning codes puts a damper on the usefulness of using a joystick for cursor positioning with CP/M programs such as WordStar. But the DT-80 does provide a useful cartridge-based 80-column terminal emulator for the Atari/ATR system. It has a number of good features and performs well. For those ATR users whose Atari does not contain an 80-column board the DT-80 is a good choice.



### Arcader's Alley By Jay Gerber

Arcader's Alley is the place for all you game players to show off your talents. Each month the highest game scores will be published, along with hints on how to play them, all submitted by you, the readers. To submit scores and hints, send them to me in a message on Aradisc, call me, or mail them to me before the 1st of each month. Remember that if you don't contribute, then you can't have your name published in the newsletter. Send all scores & hints to: Jay Gerber, 3639 N. 36th Road, Arlington, VA 22207.

Below are just a few games to start out with. If you want to add a game you are particularly good at, feel free to.

#### Top Game Scores for December, 1984 (please try not to laugh at hard)

Game	Player	Score
Pac-Man	Jay Gerber	56,420
Ms. Pac-Man	" "	71,290
Bonky Kong	" "	120,400
Bon.Kong Jr	" "	50,600
Minor 2049er	" "	37,815
Jumpman	" "	54,700
Jumpman Jr.	" "	39,000
Lode Runner	" "	Lvl 97
Seven Cities	" "	1529

**Conventions for scores are as follows:** Pac-Man, Ms. Pac-Man, Bonky Kong, Bonky Kong Jr are all played on the default difficulty level. Minor 2049er is started on Zone 1, Level 1. Jumpman is played as Grand Loop, not Randomizer option. Jumpman Jr. is played at speed 4. Lode Runner is played with the default number of lives and is recorded as highest level achieved out of 150 possible. Seven Cities of Gold is the year in which you attain a rating of Viceroy.

#### ADVENTURERS' ANSWERS

##### DARK CRYSTAL the solution



If you have seen the movie, then some of the situations will seem familiar. Doing the movie will help you in some spots, but not all of them! At the start of the game, just move in any direction. A mystic will appear. After he leaves, go east, get shale, west, west, west, north. Now you should be at Ursu, so speak Ursu and then look bowl. Go south, east, east, north, and dig, get flute. Take one step north. You will fall. Go north and cut pad, then east, listen brome, east, east, east, north, north, and use pad. After you get off the pad, go north and east. You will get caught up in some vines. Just type directions until a long arm with an eye is thrust up toward you. Speak being, moon daughters, which is the answer to the riddle.

Now, type a random direction as it won't matter because it breaks into the movie and gives you a "before Jen can react" message. This being is Anghra. She will take you to her observatory, and ask you what you want. Answer Crystal Shard. There are four different colored shards, so to find the one you want, play flate. The blue one will echo your flate, so that is the one you want. Get blue, go south.

Now the Gorthin comes, so go window, south, west, and you will fall into a bog. Now type "help" and Kira and Fizzgig will appear and help you out. Turn shall, get pouch, go shall. You then enter the pod village. Try to come around until the Gorthin arrives, then go south, west, and sit. You are now in some ancient ruins. Look wall, and read hieroglyphics. Notice the female gelfing. Go north, but before you can go, a creature will appear and say that he will help you. (He lies!) Don't follow him! (He will go South.) So, go north, west and you should be at the landstriders. Ride landstriders, go west, west, west, west, south, west, south. You will be knocked off the landstriders by the Gorthin, so type jump. While you are falling, type "grab Kira."

When you are on the ground, go west, and Fizzgig (it will ask where you want to send him. Send bers. Be another random move to waste a move, and Fizzgig will return before you can react. Use key, open barrel, and then go south, south, south, west, west, south. The chamberlain will appear and steal Kira and Fizzgig. Now go south (You will fall into a pit with Barthina). Type ran, then go hole, climb, and you will be in the Chamber of Life. Anghra will be here all wrapped up, but you can leave her as it will

#### C SQUAD

Nov 84

26

not effect the game.)

Now go east, south, west, east, east, go curtain, north, north, west, and get scepter. Go east, east, east, east and use hook. Head east, up, east and you will be on the balcony. Type jump crystal, and you will land on the crystal. You have dropped the shard, it will ask you if you want to save Kira. Answer no. Now type insert shard. At this point, the crystal is healed, but Kira is dead.

How do you heal Kira? You guessed it! Type hie Kira.

Congratulations! You have just solved the DARK CRYSTAL!

##### THE ENCHANTER the solution

When ENCHANTER begins, you are being summoned by BELBOZ to a council of the CIRCLE OF ENCHANTERS. You are told to get an end to KRILL, a nasty and powerful wizard. With your trusty spell book you are sent into the game. You find yourself at a farm in the road. You must go northeast, north, and you will be in a shack. Get the jug and the lantern and then upon the oven door and get the bread. Go south from the shack, then northeast, southeast, northeast to the shady brook. Here you fill your jug with water, head southwest, then southeast to another farm in the road. From there, go southwest, southeast to an almost deserted village. There's one place that seems to be inhabited. You head south, and run into an old crane who hands you a spell scroll, and pushes you back out the door again. This scroll has the REINOV SPELL. Use the GIMOTO SPELL to write it into your spell book. Learn the spell. Go northwest, northwest and you are back at the fork. From there head along east until you come to the outer gate or KRILL's castle. REINOV the GATE. Continue east to the inside gate. Learn the PROT, HITFUL, and REINOV spells. PROT the LANTERN, then go north twice to the tower, up the tower steps to the jewel room. There is an egg here, with all manner of little switches and doodads on it. You could open the egg by figuring out the proper sequence, but that isn't necessary.

REINOV the egg, and it will open to reveal a shredded scroll. There is no way to avoid this. Get the scroll and drop the egg. Learn REINOV once more, then go down to the tower, and east through the four error rooms to the north gate. REINOV the gate. Move north through the gate to the woods. Here you find another spell scroll. This is the KREIF spell which will repair the shredded scroll. You will only need this spell once, so you don't really have to REINOV it. Cast the KREIF spell on the shredded scroll to restore it. The spell on this scroll is ZOFNIA. GIMOTO that. Walk east to the swamp. HITFUL the frogs who will tell you how to get the CLEEN spell. GIMOTO that one too.

Return to the north gate and from there go back west through the mirror room to the tower and then from there, south till you come to the southwest tower. Go east to the south hall, then south to dungeons. Open the cell door and enter the cell. Examine the walls. A loose block! Move the block and you will be able to move east into a secret room. There is another spell scroll here, the EIEI spell. Get the scroll and SMUSTO the spell. Go west, then south and up to the south hall. Drop everything you have, and go east into the gallery. In the dark, you will see that one portrait is lit up. Move it and you will find a black candle and a black scroll. The scroll holds the GIMMO spell. Get these items and return west. Pick up your supplies. SMUSTO the GIMMO spell. About now you will feel a bit tired. Go west to the tower and up the stairs.

Get into the comfy featherbed and go to sleep. While you sleep, you have a dream. This is an indication of the location of another scroll. When morning comes, get up, then examine the bedpost. A hidden switch. Press it, and a compartment will open up, revealing the VAMM spell. Get that scroll and SMUSTO the spell. It will soon be time for you to get yourself killed, so learn the GIMMO, HITFUL, and EIEI spells.

Go down the stairs, head east until you come to the south gate. Go south from there to the meadow, southeast to the shore. You will see a giant turtle with a rainbow colored shell. Cast HITFUL on the turtle. Tell him to follow you. Return to the south gate and go east from there to the base of the southeast tower. Go up the stairs, and you will be in the engine room, which is full of all sorts of dangerous and incomprehensible machinery. Cast EIEI spell on the turtle. The speed spell will make him fast enough to dodge safely through that room into the central room, where the KULCAD spell scroll is. On his way back, he'll set off a trap, but his heavy shell will protect him. The turtle will give you the scroll. Return to the beach. The KULCAD spell is too powerful for you to SMUSTO, so you'll have to just hold on to the scroll until you need it.

Go north to the courtyard. Go east to the front of the temple, drop everything you have. Go east once more. You will be captured and put in a cell to await a sacrificial ceremony at which you will be the guest of honor.

GIMMO yourself and wait. The creatures will soon come for you. You will be offered up on an altar and a knife plunged into your heart. Because of the GIMMO spell, you won't really be dead. You now have the means of opening the jeweled box. Once you are on your feet again, stop down from the altar, and go east back to the courtyard. Cut the rope, open the box and get the MELBOR scroll. Pick up the rest of your possessions and SMUSTO the MELBOR spell. Learn MELBOR, VAMM and HITFUL. Head west, west to the inside gate and from there to the mirror rooms. You must wait

until you see the adventurer on the other side of the mirror. At that point, ZIFNIA adventurer and he will appear before you, a little bit upset. Since you have a move to spare here, MELBOR yourself, then VAMM the adventurer, who will now be very friendly towards you. He will also be looking at your inventory with covetous eyes. As soon as he's been VAMM'd, head directly east until you come to the guarded room. Your new friend will follow you along. Once at the door, tell the adventurer to open it. He will do so, and the illusions of monsters will disappear, revealing only a plain wooden door. Go north through the door into the nap room.

Here is one of those variable things in the game. There are several objects in this room, two of which the nap and the pencil, are crucial to your success. Sometimes, the adventurer will pick up one of both of them. You must get them back from him before he leaves, or you may never catch up to him again, in which case the game is lost. If the adventurer picks up something you need, tell him to give it to you, and he will. You should drop the dagger. Make sure you have the nap and the pencil, then go back to the north gate and from there, south to the library.

Examine the spheres on the floor, then the tracks in the ashes. These will lead you to a mousehole in the wall. Reach inside, and you will find the scroll with the GOMBAR spell. SMUSTO that one. While you're poking about here, you might hear guttural voices coming towards you. The MELBOR spell will protect you from any of the hairy creatures that might enter the room. There is also a dusty old book here that you might want to read, as it will help you to understand what you're doing next. From the library, return to the south hall, then go down into the dungeon, and down once more to the first translucent room. You will probably be tired, so just go to sleep. You will have the dream to find the VAMM spell book and the job: Look at the map and you will see some lettered points connected by lines. This is a magic map of the area where you are now. If you connect two adjacent points with the pencil, an opening will actually appear between those two rooms. Likewise, if you erase a line between two points, then you close off the opening between the two rooms. You can only use the eraser twice and the pencil twice before each becomes useless. You are standing at point B on the map. From there, move south, east, northeast, southeast, and you will be at point F. The point all by itself, P, is where the unseen terror currently resides, and you are about to free it. Draw a line from P to F. You will see that opening appear in the wall before your eyes, and then a very scared MELBOR will appear briefly with a warning. Move southeast twice and you will be at point P, where the GIMMO spell is. Erase the line from B to H, which will keep the terror from escaping. Also erase the line from H to V, which traps him in the room again. Pick up the GIMMO scroll, and take your way to point J. Draw a line from J to B, then wait west to B

and get your spell book. Learn the CLEENH, GOMBAR, and MELBOR spells. The GIMMO spell is too strong to be written in your book, so you'll have to carry the scroll with you. Go up twice to the south hall. MELBOR yourself, then go west to the south gate and from there north to the junction. Head east twice to the winding stairs. This is another powerful illusion. No matter how much you walk up or down, you will never get anywhere. KULCAD the stairs, and they will disappear, leaving you over a bottomless pit. Fortunately, the banister turned into a VELLUM scroll containing the IZYUK spell. IZYUK yourself and fly east into the Warlock's tower. You come face to face with KRILL himself. Before you can take care of him, you will have to get rid of a couple of his friends. When the dragon attacks, GOMBAR the dragon, and when the being attacks, CLEENH him. Now you're ready for the main event. As KRILL begins his chat, GIMMO him. He is banished forever from this plane of existence, and you have become a member of the CIRCLE OF ENCHANTERS!!

Congratulations! You have just solved THE ENCHANTER.

CCC COMPLETE COMPUTING CORPORATION  
680 EAST ROOSEVELT ROAD LOMBARD, ILLINOIS 60148

1512150000

MIKE CHECK



HELPFUL HINTS  
on solving the most of HALE  
by  
Ncutt Select

HALE from Electronic Arts is no longer a new game. Still, unlike some of the others, it has passed the test of time and remains as one of the most popular games for ATARI computers.

The object of the game is to outwit your opponents, be they live or computerized or some of each. At the end of the game, the winner is the one with the most 'earthly' goods and the greatest amount of money and land.

Here are five hints to help you win the game:

1. When you are on the purchasing end, be sure to take your opponents out! Never reveal your true intent till just before the sale is completed.
2. Cavest emperor! Buyer beware. Just as you will be attempting to steal from them, so will they be trying to steal from you!
3. Take the Packer if you're planning to farm. Take the Donzoid for seithers. Take the Muamimid otherwise.
4. No matter what your intent, get rid of the farm land you can. Don't let them get it if you can help it.
5. In the last month, change everything you have to the highest commodity, either 'seithers' or trybites, depending on the level.

One of the best features of the game is the ability to choose the level of play. Once you have become good at the lower levels, you may play at the tournament level, making the game a constant challenge.

Perhaps it would also be a good sixth tip to suggest you play against human opponents whenever possible. They are always easier to fool! (I know. I usually play against Lisa!)



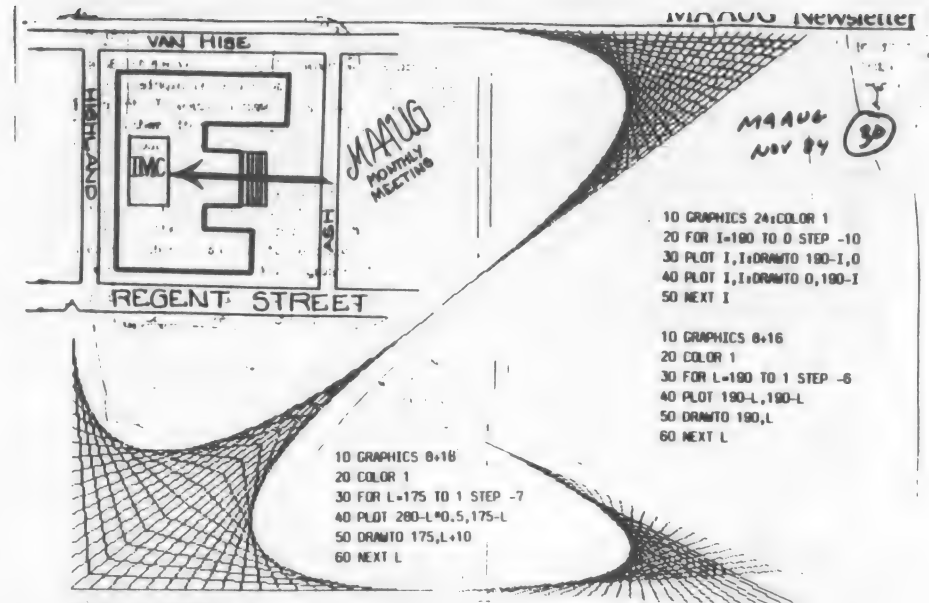


# assembly language

29

## Decimal and Hex Codes for Instruction Set NUMERICAL

DEC	HX	CHAR	opc	Alform	DEC	HX	CHAR	opc	Alform	DEC	HX	CHAR	opc	Alform
0	00		BRK		89	59		EOR	aaaa,Y	174	AE		LDX	aaaa
1	01		ORA	(aa,X)	93	5D		EOR	aaaa,X	176	80		BCS	aa
5	05		ORA	aa	94	5E		LSR	aaaa,X	177	81		LDA	(aa),Y
6	06		ASL	aa	96	60		RTS		180	84		LDY	aa,X
8	08		PHP		97	61		ADC	(aa,X)	181	85		LDA	aa,X
9	09		ORA	*nn	101	65		ADC	aa	182	86		LDX	aa,Y
10	0A		ASL	A	102	66		ROR	aa	184	88		CLV	
13	0D		ORA	aaaa	104	68		PLA		185	89		LDA	aaaa,Y
14	0E		ASL	aaaa	105	69		ADC	*nn	186	8A		TSX	
16	10		BPL	aa	106	6A		ROR	A	188	8C		LDY	aaaa,Y
17	11		ORA	(aa),Y	108	6C		JMP	(aaaa)	189	8D		LDA	aaaa,X
21	15		ORA	aa,X	109	6D		ADC	aaaa	190	8E		LDX	aaaa,Y
22	16		ASL	aa,Y	110	6E		ROR	aaaa	192	C0		CPY	*nn
24	18		CLC		112	70		BVS	aa	193	C1		CMP	(aa,X)
25	19		ORA	aaaa,Y	113	71		ADC	(aa),Y	196	C4		CPY	aa
29	1D		ORA	aaaa,X	117	75		ADC	aa,X	197	C5		CMP	aa
30	1E		ASL	aaaa,X	118	76		ROR	aa,X	198	C6		DEC	aa
32	20		JSR	aaaa	120	78		SEI		200	C8		INY	
33	21		AND	(aa,X)	121	79		ADC	aaaa,Y	201	C9		CMP	*nn
36	24		BIT	aa	125	7D		ADC	aaaa,X	202	CA		DEX	
37	25		AND	aa	126	7E		ROR	aaaa,X	204	CC		CPY	aaaa
38	26		ROL	aa	129	81		STA	(aa,X)	205	CD		CMP	aaaa
40	28		PLP		132	84		STY	aa	206	CE		DEC	aaaa
41	29		AND	*nn	133	85		STA	aa	208	D0		BNE	aa
42	2A		ROL	A	134	86		STX	aa	209	D1		CMP	(aa),Y
44	2C		BIT	aaaa	136	88		DEY		213	D5		CMP	aa,X
45	2D		AND	aaaa	138	8A		TXA		214	D6		DEC	aa,X
46	2E		ROL	aaaa	140	8C		STY	aaaa	216	D8		CLD	
48	30		BMI	aa	141	8D		STA	aaaa	217	D9		CMP	aaaa,Y
49	31		AND	(aa),Y	142	8E		STX	aaaa	221	DD		CMP	aaaa,X
53	35		AND	aa,X	144	90		BCC	aa	222	DE		DEC	aaaa,X
54	36		ROL	aa,X	145	91		STA	(aa),Y	224	E0		CPX	*nn
56	38		SEC		148	94		STY	aa,X	225	E1		SBC	(aa,X)
57	39		AND	aaaa,Y	149	95		STA	aa,X	228	E4		CPX	aa
61	3D		AND	aaaa,X	150	96		STX	aa,Y	229	E5		SBC	aa
62	3E		ROL	aaaa,X	152	98		TYA		230	E6		INC	aa
64	40		RTI		153	99		STA	aaaa,Y	232	E8		INX	
65	41		EOR	(aa,X)	154	9A		TXS		233	E9		SBC	*nn
69	45		EOR	aa	157	9D		STA	aaaa,X	234	EA		NOP	
70	46		LSR	aa	160	A0		LDY	*nn	236	EC		CPX	aaaa
72	48		PHA		161	A1		LDA	(aa,X)	237	ED		SBC	aaaa
73	49		EOR	*nn	162	A2		LDX	*nn	238	EE		INC	aaaa
74	4A		LSR	A	164	A4		LDY	aa	240	F0		BEQ	aa
76	4C		JMP	aaaa	165	A5		LDA	aa	241	F1		SBC	(aa),Y
77	4D		EOR	aaaa	166	A6		LDX	aa	245	F5		SBC	aa,X
78	4E		LSR	aaaa	168	A8		TAY		246	F6		INC	aa,X
80	50		BVC	aa	169	A9		LDA	*nn	248	F8		SED	
81	51		EOR	(aa),Y	170	AA		TAX		249	F9		SBC	aaaa,Y
85	55		EOR	aa,X	172	AC		LDY	aaaa	253	FD		SBC	aaaa,X
86	56		LSR	aa,X	173	AD		LDA	aaaa	254	FE		INC	aaaa,X
88	58		CLI											



## ENHANCED DENSITY IN DOS 2.0 FOR THE ATARI 1050, RAM AND DISK DRIVES by John Reuschlein

Here are some interesting changes which can be made to DOS (excepting for you 810 owners).

The disk in normal single density contains 720 sectors. Only 707 are actually available for storage of your programs and files. The enhanced density version of DOS contains 963 sectors for program and file storage. That's a difference of 256 sectors. The remaining sectors are used by the disk operating system (DOS). Sector 360 is utilized by the directory or so-called 'Volume Table of Contents' (VTC). A status of the individual files is stored here as well. This sector can be changed using BASIC to increase the actual amount of storage space available on the disk for programs and files. If we increase the amount of space and stay with DOS 2.0, we can increase the efficiency of the system (DOS 3.0 is slower and not as easy or convenient to use).

DOS 579 is the program which we will alter in order to force more data to be stored on the disk. Be sure you have the drive booted with DOS available before you start. The Table below lists the several

POKE locations you will have to use:

- POKE 1974,n (where 'n' means 33 for single density and 34 is for medium density)
- POKE 3363,n (where 'n' is the same as above)
- POKE 3426,195 (sector count low byte)
- POKE 3434,3 (sector count high byte)
- POKE 3443,8 (start of sector record)
- POKE 3450,128 (last byte of sector record)
- POKE 3458,8 (same as location 3443)
- POKE 3480,51 (points to the directory and reserves 8 file sectors)
- POKE 4334,8 (same as 3443)
- POKE 4350,5 (last byte in the sector record where the sector count data is stored)
- POKE 4382,128 (same as location 3450)

After the above pokeys are typed in directly or run from a program, go into the DOS. Format a disk and write the DOS files to the newly formatted disk using standard option 'H'. If everything goes smoothly you should end up with an extra 256 sectors at your disposal in DOS 2.0.

## Make Your Own EPROM Cartridge

By Bekki Kusbe  
Robert Bodine

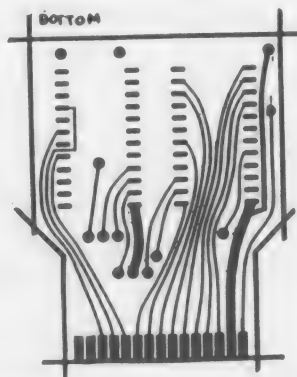
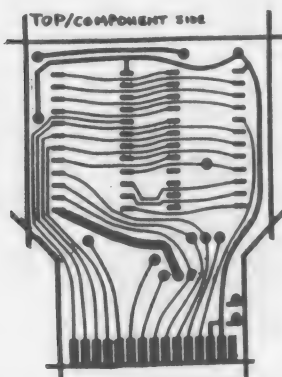
Why make your own cartridge? There can be no more than two good reasons.

#1 You just want to learn the nuances of EPROM interfacing and timing.  
#2 You want to market the super-game program you have written in a cartridge and can't afford mask programmed ROMs.

Either way there are some tidbits which may prove to be useful at some point in your effort. The ATARI ROM board is about 2.6" high by 2.1" wide top edge and 1.580" wide card edge at bottom. There is space for a 30 pin connection, 15 on each side. The connector traces are 0.1" center to center. They are numbered 1-15 on the component side (facing you) and A-R (letters I, O, Q are not used due to possible confusion with numbers) on the opposite side. The card is designed to be populated by two 4K by 8 bit ROMs. It takes 12 lines to address a 4K byte space (anyone remember POP-12's). So the pins A0-A11 of the ROMs are directly connected to the A0-A11 lines of the address bus. Address line A12 extends address space to 8K bytes by selecting the lower 4K ROM when 0 and the upper 4K ROM when 1. Two more criteria for selection are the lines which come in on pins 1 and 12. These lines are outputs of an address decoder on the motherboard which get selected in 32K-40K range for the right slot (800 only) and 40K-48K for the left slot. Following is a table of the pin-outs of the cartridge board.

NOV 84 (31)

Name	Label	Trace
Address Bus	A0	5
	A1	A
	A2	3
	A3	2
	A4	C
	A5	D
	A6	E
	A7	F
	A8	H
	A9	J
	A10	P
	A11	N
	A12	K
Data Bus	D0	10
	D1	9
	D2	8
	D3	L
	D4	6
	D5	7
	D6	11
	D7	M
Chip Select	cs	K
	cs	I
Power 5 volts	Vcc	13
	Vcc	14
	GND	8



## MAAUG Newsletter

The ATARI ROMs are proprietary devices and by design have a slightly different pin-out than generic EPROMs. So one cannot simply plug in a 2732 EPROM in the ATARI card and expect it to run. Following are the differences between the ATARI ROMs and the popular 2732 and 2532 EPROMs even though they are all 24 pin devices..

Pin	ATARI	2732	2532
18	A11	CS	A11
21	CS	A11	SV+

The ATARI and the 2732 both need two chip select lines (active low). The 2532 has only one CS line. Because lines K and 1812 decide which EPROM is to be selected it is easier to interface the 2732 EPROMs. The 2532 would need additional logic to decode these lines. Included here are the PC-card layouts to make a 2732 EPROM cartridge. The caveat is that these have not been tried yet. Sometime in the future when the PC-lab to which we have access goes on line again a debugged version will be available at a nominal cost. Contact the authors if you are interested.  
(608-263-1564)

## The Epson FX-80

by Stephen McMillan

In keeping with our upcoming meeting, centering on hardware, I've written this overview of the Epson.

Before buying the Epson I had been worried that the FX-80 didn't have a thing called 'Grafrax', but after using the printer for a while I found that it did indeed have this feature. Grafrax, is a part of the Read Only Memory (ROM), and affords the FX-80 compatibility with the many outstanding printer graphics dump programs, both public and private domain. It also allows you to access graphics as fine as 480 x 480.

The printer has sixteen different print mode combinations that can be selected by number in BASIC, along with a couple of other of commands. You can get any style you really need. A few of the styles you can use are: Roman, Italics, compressed, expanded, boldface.

A 2K buffer is available, or you can use the extra memory for your own character set. You'll never want to go back to 'bufferless' word processing after using one!

But, probably the best feature of this printer is the paper handling features; such as, forward and reverse paper feeds, and the most useful, reverse

line feed. Reverse line feed lets you print foreign language punctuation marks, and special marks, for some languages.

Overall the Epson FX-80 is in a much higher class than any other printer in its category; such as, the Star 10x, or the OKIDATA 92, etc.

## The Gemini 10-X

Star Micronics Inc. \$240.00

by Dave Divine

The Gemini (like a gem) is a powerful printer. It has 8 foreign character sets for: USA, England, Germany, Denmark, France, Sweden, Italy, Spain. It has options including italic, boldface, double strike, with underline, subscript, and enlarged lettering. It prints 120 characters per second at 10 characters per inch.

The Gemini has great graphics capabilities. It will print 60 by 72 dots per square inch in low res. 120 by 144 in medium res and 240 by 144 in high res. It allows you to download custom character sets and prints control characters. The BASIC column of this newsletter and the images in last month's graphics section were done on my Gemini printer. It shows the real capabilities of this printer.

The Gemini uses a standard typewriter ribbon and has traction and pin feed; options; uses a standard RS232 interface; is similar in shape and size to the Epson, and is compatible in instances in which the Epson is compatible. But the best break is that it is only \$240.

I use my Gemini printer all the time and never have had any problem with it. I have it interfaced with an Ape-Face, use the screendump programs from ANALOG magazine which I have mentioned, and in combination with BANK Street Writer, I have all the power and capability I need for school papers, personal letters, art and graphics and any other needs that have come up. It's a really, great printer!

## THE BASIC COLUMN

Quickie Graphics  
by Dave Divine

Here, are three short basic programs which give you pretty pictures on the screen. By adding the screen dump program you can print them out in graphics mode 8. Try chaining some of the values for I and L to get different effects.

by Kenneth J. Pietrucha - JACG

The following program was written to study the repeating number sequence in proper fractions to as many places as desired. No attempt has been made to conserve memory. It is a simple brute force approach. As given, the program will run to 500 decimal places. The limiting factor is the size of the DIMENSION statements and the calculator loop in 130.

```

10 REM INFINITE PRECISION DIVISION
20 REM by Kenneth J. Pietrucha
30 REM J.A.C.G.
40 REM September 2, 1984
50 PRINT
60 REM USE WITH PROPER FRACTIONS ONLY
70 PRINT
80 PRINT CHR$(125)
100 DIM A(500),N(500),T(500),R(500)
110 PRINT "INPUT NUMERATOR:INPUT A"
120 PRINT "INPUT DENOMINATOR:INPUT B"
130 PRINT CHR$(125)
140 PRINT A;"/";B;"=";",";
150 FOR I=1 TO 500
160 R(I)=A
170 N(I)=R(I)*10
180 A(I)=INT(N(I)/B)
190 PRINT A(I);
200 T(I)=A(I)-B
210 NEXT I

```

The program is simple and actually goes through the same steps that you would go through if you were solving the problem by hand using long division. The first line, of course, tells the computer what to do. In other words, let's take the fraction 7/22. After entering the numbers we begin at line 125 by printing the decimal point. We assign R(1) the value of 7 at line 140 and in line 150 we multiply by 10, which is the same as adding a zero after the 7. If you are doing this with a pencil, you now have 22 which you are dividing into 70. The integer part of the answer obtained by using a calculator is 3, which is exactly what we get in line 160. Line 170 prints the 3. Now if you follow me we take 3 and multiply it by 22 in line 180. This gives us 66 and we then write 66 below 70 and then subtract. The 4 we get is the difference of 4 which we assign to the variable A in line 190. The loop continues back at line 140 and we go through the same steps again and again.



33

One last thing. If you want to print the results on a printer, POKE 838,166: POKE 839,238 to send the run to the printer. You may have to wait until the printer buffer fills before it starts printing. Do a POKE 838,163:POKE 839,246 to send the run back to the screen.

## by Michael Stoen

Dynacorp, Inc  
1427 Monroe Avenue  
Rochester, NY 14618  
(716) 442-8960

- 1) "Microcomputer Bond Program"  
Bond analysis; \$39.95
- 2) "IRMA"  
Portfolio management; \$49.95
- 3) "Tax Optimizer"  
Tax planning & prep.; \$39.95
- 4) "Microcomputer Stock Program"  
Technical analysis; \$39.95
- 5) "MYINDEX"  
Technical analysis; \$29.95
- 6) "Stockaid"  
Technical analysis; \$29.95

Software Models  
P.O.Box 1029  
23913 Bowl Road  
Crestline, CA 92325  
(714) 338-1238

- 1) "Finance Models"  
Financial planning; \$39.95
- 2) "Home Models"  
Financial planning; \$39.95
- 3) "Real Estate"  
Real estate, natch; \$39.95

**Advanced Financial Planning**  
20922 Paseo Olea  
El Toro, CA 92630  
(714) 855-1578

- 1) "Life Insurance Planning"  
Financial planning; \$29.95
- 2) "Retirement Planning"  
Financial planning; \$29.95

Atari Program Exchange  
P.O.Box 3705  
Santa Clara, CA 95055  
(408) 727-5403 (good luck!)

- 1) "Strategic Financial Ratio Analysis"  
Fundamental analysis; \$24.95
- 2) "Stock Management"  
Portfolio management; \$24.95
- 3) "Real Estate Cash Flow Analysis"  
Real estate; \$24.95

Continental Software  
11223 South Hindry Avenue  
Los Angeles, CA 90045  
(213) 417-8031

- 1) "The Tax Advantage"  
Tax preparation: \$69.95

It wasn't on the list, but Continental Software also makes "The House Accountant", which I've looked at, and it seems to be a pretty good system for handling your house finances. Perhaps they don't count that as investment software? The only program on the list that I've used myself is "The Tax Advantage", and I must report that it certainly makes doing my tax return a lot easier. I know of another one: the "Dow Jones Investment Evaluator". There was only one problem with it; it didn't work! Dow Jones has since discontinued offering it for the Atari.

All in all, not an impressive list. We can only hope that things will improve in the future. I still find the most useful bit of investment software I have is Visicalc!

If you want to learn more about the AAMI, and see what they have to offer, you can write them at:

AAHI  
P.O. Box 1384  
Princeton, NJ 08542  
(609) 737-3972

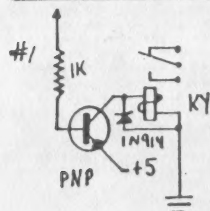
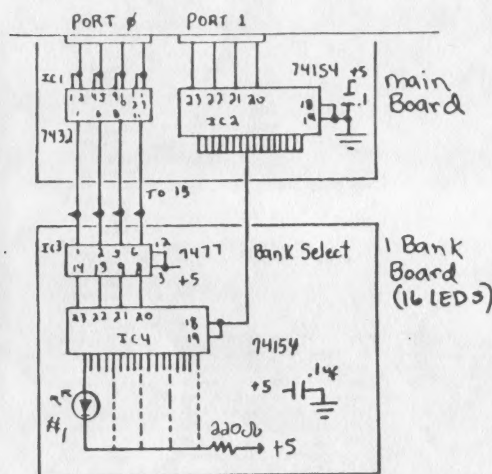
A.B.A.C.U.S.- For those of you headed that way, the Atari Bay Area Computer Users' Society meets the first Saturday of each month from 10am to 2pm at Building C, Room 300, Fort Mason Center, in San Francisco. Their address is P.O. Box 1823, Mill Valley, CA 94941, and can be reached at (415) 753-8403. Their BBS is (415) 587-0662.



# LED CONTROL BY PORT

The Hobby Shop  
by Rick Detleisen

This time I will present a circuit to light LEDs with an output port. As the circuit is fairly straight forward, I won't explain it in detail. Refer to the last article for more info. Part numbers are listed with the schematic. The bank select line connects to ONE of the control lines of IC2 (unless you want more than one board responding to the data - i.e. bank board 0 to control 0, bank board 1 to control 1, etc.). The latch (IC1) is optional, jumper the lines if you want a momentary display. Its function is to keep the data on the input lines of IC4, to allow the computer to do other things. Once the latch has the data, the computer can be turned off while the display stays the same (until its power is turned off).



Also, note that only one output is on at a time. Other circuits could provide individual control over each output.

Obviously, lighting LEDs may be fun, but it is not very useful (or cheap!). Note #1 is the LED or the relay control circuit. The relay can control almost any external device (again only one per bank board). With volume (voltage) changing as the control number increases, if desired. One example (with the latch jumpered) is controlling an AC sonic home control system--let your computer push the buttons, or now about letting your computer operate your house lights in your USBUL pattern while your on a trip? Oh, before I forget, IC1 is a buffer IC, to allow if other boards to "plug in" to the joystick port without overloading it.

Next time, I will get down to the circuits for reading switches--reading 25b switches is fairly easy, reading more starts to get complex but it too can be done.

AACB  
12-84

35

JACK NOV 84

36

What Language Does Your Atari Speak?  
A Guide to Programming Languages  
Available for the Atari Computer  
By Arthur Leyenberger - JACK

Copyright (c) 1983 by Arthur Leyenberger

Welcome back to my continuing series of articles on programming languages. I hope you enjoyed last month's discussions of Machine Language, Assembler, Disk Operating Systems, and Basic. This month I'll talk about Atari Microsoft Basic and get into some of the more esoteric, but useful, languages.

## TOP-DOS

Last month I talked about Disk Operating Systems but failed to mention two important and significant DOSs: TOP-DOS (used to be called DOS-Mod) and MYDOS.

A useful DOS utility and one that I would recommend to anyone who does a lot of file manipulation is TOP-DOS from Eclipse software. TOP-DOS is completely compatible with Atari DOS 2.0S and contains a number of enhancements. Although TOP-DOS has many, many more features than Atari DOS 2.0, the memory-resident portion of TOP-DOS is the same size as that of Atari DOS.

How would you like to be able to use full screen editing when using DOS, just like you do when using Atari Basic? You got it. How about being able to see more of what you are doing on the screen? No problem, TOP-DOS gives you more than half the screen since its compressed menu takes up less room. Another aid to let you know what is going on is the minimum of screen clearing that occurs.

If this was all that TOP-DOS allowed me to do I would say, "ok, but so what?". There's more. Commands that used to take several lines and require answering prompts can now all be put on one line. The wildcard operators now work (as they should) with COPY, DELETE and RENAME. And get this, TOP-DOS lets you create command files which may contain a sequence of commands that will be executed in one operation. Finally, TOP-DOS fixes many of the bugs in Atari DOS such as the RS-232 handler being destroyed on RESET and typical MEM.SAV problems.

## MYDOS

MYDOS is an alternative Disk Operating System (DOS) for the Atari computers. Produced by SWP, Inc., the makers of the ATRODOS, MYDOS is intended primarily for their machine but will also work with Percom and other higher density disk drives (double density and double sided). The strength of this DOS lies in its power and ease of use. The program is not necessarily user friendly - you must know what you are doing or you could easily trash an entire disk - but it is easy to

use. Like TOP-DOS, multi-line, multi-keystroke commands can all be entered on one line, with fewer keystrokes and much more clearly.

MYDOS supports upper and lower case file names too. That is another "useful" but not necessarily "friendly" feature. Although you can pre-define the density of the disk drive (and disk) that you are using, MYDOS is smart enough to be able to read a disk in a format different from what you specified. When copying an entire disk with the "J" command, MYDOS is nice enough to automatically format the disk for you unless you say otherwise.

## Atari Microsoft Basic

Atari Microsoft Basic comes in two flavors - a disk based version and a cartridge/disk version (Atari Microsoft Basic II). Both versions contain the same capabilities but Atari Microsoft Basic II is preferred because the cartridge is a more durable medium for the language than an uncopyable disk.

There are a number of features implemented quite differently in Atari Microsoft Basic (AMSB) than in Atari Basic or Basic XL. AMSB is a superset of Microsoft Basic and contains Atari-specific features for sound and graphics. There is no syntax checking during line entry so errors will not become apparent until the program is run.

Unlike Atari Basic, you can choose the precision of numeric variables. Single, double or integer variables are allowed. Math functions are performed in the interpreter rather than in the floating point package in the OS ROM which increases the speed of calculations.

One of AMSB's great advantages over Atari Basic and Basic XL is its ability to accept user-defined functions. This essentially lets the user make up their own Basic commands. Although there are no commands for joystick or paddle reading, they can easily be implemented with DEFINED functions or PEEKs.

Perhaps the most significant difference between AMSB and other Atari Basics is the way in which strings are handled. In Atari Basic strings are one-dimensional, must be DIMensioned and can be as long as memory allows.

AMSB does not require one-dimensional strings to be DIMensioned, only allows a maximum string length of 255 characters and permits true string arrays. Other useful string functions are also included, such as: true concatenation, left, mid and right substrings and substring search.

There are quite a few advanced "housekeeping" commands in AMSB like automatic line numbering, line renumbering and line deletion. There are direct commands that interact with DOS such as KILL (delete a file), NAME (rename a file) and LOCK/UNLOCK. Other useful advanced

features allow the ability to trace the execution of a program and pass the values of variables from one program to another. Additional or modified statements include WAIT (for halting a program for a specific amount of time), IF...THEN...ELSE (for better program testing) and the combining of PLOT and DRAWTO into one command - PLOT TO.

#### PASCAL

Pascal was invented by Kathleen Jensen and Niklaus Wirth as a tool for teaching ALGOL (Algorithmic Language) and to demonstrate the principals of a structured language. Pascal is an easy language to learn and is suitable for defining the data structures needed for problem solutions. The language was named for Blaise Pascal, the French mathematician who invented one of the first mechanical computing devices.

Pascal allows data types such as integer, real, and Boolean, but gives the programmer the freedom to define new data types. New functions and procedures may be defined and character data and strings may be manipulated. It is a compiled language, but it does not usually compile into machine code. Instead, it compiles into an intermediate pseudo-code called p-code. The p-code is then saved and at run-time, the file is interpreted into the machine code of the computer.

There are two implementations of Pascal for the Atari computer. One is available from the Atari Program Exchange (if you can still get it). It is based upon the standard PASCAL and is very similar to UCSD PASCAL. There are some minor differences involving the incorporation of graphics and sound capabilities. Unfortunately, its use is limited because it requires two disk drives and is not supported officially by Atari.

#### Draper PASCAL

The other version of PASCAL for the Atari computer is called Draper PASCAL. Many folks have knocked Draper PASCAL because it does not conform to the official PASCAL standard. It is not a full implementation of the language (either ISO or UCSD standards) and includes an abbreviated range of data types, a lack of number formatting and limitations on parameter passing. Now that I have told you what it is not, and what it can do, let me tell you what it can do.

For one thing, Draper PASCAL is close enough to the real thing to make it suitable for learning the language or programming applications on the Atari. The language has machine-specific features such as disk management, I/O, string manipulation, BASIC-type graphics and sound and ability to read joysticks. There is also a CALL procedure which allows you to easily access your own machine-language subroutines.

I am not a fluent PASCAL programmer by any stretch of the imagination but I am familiar with the language and can write fairly simple programs. I was able to have a small PASCAL program written, debugged and running using Draper PASCAL in under 20 minutes from first opening the package.

The conclusion is simple. If you must have PASCAL for the Atari computer then Draper PASCAL is the only game in town. It's relative power and ease of use will make you feel like you are coding an application for one of the "big rigs".

#### C Language

The language called C was designed at Bell Laboratories to exist within the UNIX Operating System. It is a structured language with some similarities to Pascal. However, the entire program structure is built through the use of functions called modules. There are no Print or Read statements and input/output is also done through use of modules. I/O structure is performed through the I/O structure of whatever operating system it is implemented on.

There are no line numbers in C. A program is written using an editor, then it is compiled and a linker is used to link all of the program modules together. A program starts with the name of the function, then a left bracket to start function definition. This consists of compound statements enclosed within two square brackets. Statements may be nested to any depth and are treated just like simple statements. There are libraries of standard functions and those functions previously defined by the user. Global and local variables, arguments for the functions and expressions used to calculate and store data are allowed. C can call machine-language routines when needed in addition to any of the user-defined or standard functions.

C is available in two versions for the Atari: Tiny C and Deep Blue C. Tiny C (now called C/65) is available from Optimized System Software and Deep Blue C is available from API (AFX-20166).

#### See You Next Time

That's it for this month. Next month I will wrap up the series by discussing LISP, FORTH, Logo and Action!. Until then, may all your subroutines compile.

**HAVE YOU RENEWED  
YOUR MEMBERSHIP?**

**CHECK YOUR MAILING LABEL  
FOR MEMBERSHIP EXPIRATION DATE**

\*\*\*\*\*

39

**What Language Does Your Atari Speak?**  
A Guide to Programming Languages  
Available for the Atari Computer  
By Arthur Leyenberger - JACC

Copyright (c) 1983 by Arthur Leyenberger

Welcome back to this, the last article in the series of articles concerning programming languages. I hope you have enjoyed the previous months discussions of Machine Language, Assembler, Disk Operating Systems, Basic, PASCAL and C. More importantly, I hope you have learned something. This month I'll talk about LISP, FORTH, Pilot, LOGO and Action.

#### LISP (List Processing Language)

LISP is the language used in artificial intelligence research, and has since been implemented on many computers, including the Atari. There are two kinds of words in LISP, atoms and lists. Atoms are the basic entities of LISP. Any combination of alphabetic characters with any of the ten digits is an atom, as long as it starts with a letter. A list is the second type of word in LISP, and it is built up from atoms and other lists. A list consists of a left parenthesis followed by any number of atoms and lists, terminated with a right parenthesis. The language has functions, variables, and arithmetic operators, but it looks strange to Basic programmers because all the arithmetic operations are in Reverse Polish Notation (RPN).

We are all familiar with Algebraic Notation since this is how we initially learned to add, subtract, divide and multiply. To multiply two numbers, say 5 and 7, we would write, "5\*7". In Reverse Polish Notation, the operator comes after the numbers. Therefore, we would write, "5 7 \*". It is called "Polish" Notation because of the Polish mathematician, Lukasiewicz. It is called "Reverse" because, unlike Lukasiewicz's original logic, the operator comes last, not first.

A LISP sentence looks like a list, but it carries meaning and it is actually an elementary program. All LISP functions can be applied to arguments. The LISP language has many built-in functions, and the programmer can create his or her own functions. There is only one currently available version of LISP for the Atari computer. It is available from Datamost.

#### FORTH

Forth is not an easy language to learn since it is different from anything we are used to. In addition, it does its calculations in Reverse Polish Notation (RPN). It is sometimes called the unfinished language because the programmer has almost unlimited freedom to create new words or functions. Everything in Forth is a word. It is not very good as a number

cruncher, but it can link to subroutines in other languages for the more complex math. The programmer defines new words by using old ones. Very little original work is needed to write a new program because the system uses all of the work that was done before, as if it was part of the language.

It is a structured language with no GOTOS or labels for statements. Since the program is later compiled into machine-readable code, very little space is needed in memory. A full Forth can fit into a 16K machine and still have room for programs. In addition, it is inexpensive. The Forth Interest Group (FIG) has made versions available for almost every computer including the Atari.

FIG-Forth is available from the Atari Program Exchange (AFX-20029) and Val-Forth is available from Valpar International. Val-Forth uses separate modules that contain Atari-specific features such as Player-Missile graphics, Display-List Interrupts and sound capability. These modules are inexpensive and may be purchased separately, as you need them.

**PILOT (Programmed Inquiry Learning Or Teaching)**

PILOT was the first computer language dedicated to computer-aided instruction, and has been implemented on many computers. This interactive language allows a peer response and give advice or comment based upon that response. PILOT instructions are divided into four categories:

1) Single-letter core instructions which are standard for all of the versions of PILOT. Thus, the programs are portable from machine till appear on the screen.

2) Instructions that set various kinds of parameters related to the computer such as output ports, display speed, or memory location.

3) File system instructions relating to storing and retrieving programs and data on tape or disk.

The Atari Version of PILOT includes Turtle Graphics (see below). PILOT is probably the easiest language to learn for the new computer user.

#### LOGO

The LOGO language is easy to use yet sophisticated enough for higher instruction. LOGO uses basic modes called sprites and turtles. The sprites are forms that the user creates that can move around the screen at any speed the user selects. The turtle is a figure that the user can interact with, moving it over the screen, coloring it, and making it draw or erase lines. Instructions to the turtle can be either absolute or relative. For example, from the initial position, GO 10 (relative) and GOTO 1, 10 (absolute) are equivalent commands. It is especially easy for children to learn the relative commands

JACC  
DEC 84 38

since the child can mimic the responses of the turtle by acting out the commands themselves.

More complex functions (programs) can be created that children can interact with through simple keyboard responses. Children may learn color, direction, letters, words, and sounds through this medium and usually find it fun. It also introduces them to the use of the computer.

Atari LOGO is a good implementation which, among other useful features, contains the capability for manipulating and programming four turtles.

#### ACTION

Action is a new language from Optimized Systems Software (OSS). It is somewhat of a cross between Assembler and C. However, it's similarity to Basic will allow experienced Basic programmers to convert to this language with few problems. Action is a proprietary product, and is only available from OSS.

The Action system is composed of an editor, the language itself, a compiler and a monitor. The editor is used when the system is first booted up from the cartridge. It allows you to create and modify Action programs. The editor contains two text windows that can be moved throughout the program. In addition to the scrolling window, the editor contains search and replace, delete lines and move blocks of lines features.

The compiler is the heart of the Action language system. After the program has been created, it must be compiled using the Action compiler. This process transforms the relatively english-like Action program into machine language. "Include" statements used with the compiler allow several separate program modules to be combined into one executable file. For example, this would be useful for including previously written general purpose subroutines within a particular program. The include command can be nested up to six levels. This means that an included subroutine can include another subroutine which can include another, etc.

The monitor is the control section of the system. From there you call the editor, the compiler, run a program or exit from the Action language. Access to DOS is also gained through the monitor.

The Action language itself is very structured. Like C, procedures or modules are written separately then grouped together into a program. Expressions may contain arithmetic, bit-wise or relational operators. Many powerful statements are provided such as IF...THEN...ELSEIF...ELSE. Other useful statements include WHILE, UNTIL and DO loops.

The Action language system also contains a library of useful routines. These routines include capability for input, output, I/O support and system functions. The I/O support routines are particularly useful for the Atari programmer because functions such as Open, Close, XIO, Note, Point, Color, Sound, Stick and Paddle are provided.

The programmer experienced to the relatively unstructured world of Basic may have some difficulty adjusting to the very structured Action. However, after a few hours with the language, the light is seen at the end of the tunnel and it all seems natural. Indeed, it may be more difficult going back to Basic after experiencing the speed and power of the Action language.

#### CONCLUSION

I have covered a lot of ground in this series of articles. Ten different languages have been discussed, some with several dialects. A total of 18 different ways to "talk" to your Atari computer.

Each of these languages have their strengths and weaknesses. Some are better for scientific applications while others are better for the first-time computer user. Other differences such as speed, graphics capability and ease of use are also apparent. Table 2 provides a brief description of the capabilities and application of each of the languages.

As you have seen, there are quite a few languages available for the Atari computer. This clearly makes the Atari computer competitive with other, more costly machines. In addition to having the greatest game computer available, Atari owners also have a serious tool at their disposal, which can be a workhorse for application programming.

I hope that I was able to interest you in learning at least one new language for programming your future applications.



It was bound to happen—they're beginning to think the jewelry category.

\*\*\*\*\*

#### TELECOM

By Lawrence Moriano - JAGC

A few months ago I started a discussion about CompuServe and the many features it offers the home user. This month I will explore Services For Professions, along with a discussion on the Electronic Mail section.

The Service For Professions category of CompuServe's main function is to help specific professions. The home user will find this section to be quite useful if you are looking to increase your knowledge about a profession you are not accustomed to. This category has seven sections which I will briefly discuss by highlighting its main features.

1. Aviation
2. Communications/Data Processing
3. Engineering/Technical
4. Environmental
5. Legal
6. Medical
7. Jewelers

In the Aviation section you can gather information on aviation weather reports, flight planning, the electronic addition of the official Airline Guide as well as many other exciting features.

The Data Processing section is geared towards the data processing field. There is a listing of job opportunities along with a monthly news letter gathered from data processing companies.

If you are looking for legal advice for your company or personal needs, then the legal section can be important tool for you to utilize.

If you are thinking about buying diamonds or other types of jewelry for investment or personal needs then the jewelry category will come in handy. This section will give the investor an insight to the changing jewelry market along with many other items of interest.

Engineers will find a large database of literature available on science, engineering, and environmental related topics. You can receive information and order documents which are provided by the many suppliers who are a major part of this database.

Doctors of medicine, dentistry and the veterinarian areas can obtain news as well as information pertaining to their own specific needs.

A very important part of any business is the ability to communicate quickly with associates and corporations in your own community as well as across the county. EMAIL can provide this service to anyone who is a subscriber of ComuServe.

With EMAIL you can send messages as well as receive messages from other subscribers. There is a 4000 character, 20 message limit before your files will be filled. An important aspect to consider when using EMAIL is the ability to communicate to business associations all over the country for the price of a local phone call! Pretty sneaky. Old Ma Bell will soon be missing the high monthly long distance phone bills.

I have briefly highlighted the Service For Profession category. Might I emphasize briefly I have barely scratched the surface of this very important part of the CompuServe system. So get on line and see for yourself.

It's hard to believe that the holiday season is right at our door steps. I would like to take this opportunity to wish everyone a happy holiday season. May the upcoming new year be healthy and prosperous to you and all the members of your families.

UNTIL NEXT YEAR  
MERRY CHRISTMAS  
AND A HAPPY NEW YEAR



COMPUTER WHIZ...